

# Using the SBO\_SP\_TransactionNotification Stored Procedure



## Applies to:

SAP Business One, SDK, SBO\_SP\_TransactionNotification stored procedure

## Summary

This article explains how to use the SBO\_SP\_TransactionNotification stored procedure to receive notification of a data-driven event in SAP® Business One. You can also download the code samples.

**Author:** James Kofalt

**Company:** SAP

**Created on:** March 2004

**Last update:** May 2009 by SAP B1 Solution Architects

## Table of Contents

Motivation.....	3
The SBO_SP_TransactionNotification in detail.....	3
Important Note for 2007 version .....	5
Adding Code to the Stored Procedure.....	5
Related Contents .....	6
Copyright.....	7

## Motivation

Partners frequently ask me whether there is a way to receive notification of a data-driven event in SAP Business One. In other words, developers need a way to trigger events in their application based on something that happens in SAP Business One. For events that occur within the SAP Business One client application, it is as simple as using the User Interface API (UI-API) to trap events and respond to them. However, for events where the client application is not involved, for example when someone inserts a document through the Data Interface API (DI-API), another mechanism is required.

The mechanism for receiving notification of data-driven events is then to locate some code inside the existing B1 stored procedure called SBO\_SP\_TransactionNotification.

Please note that adding SQL triggers or Stored Procedures at the database level is not permitted (cf. [SAP Business One Add-On Solution Certification](#) document) – in addition triggers will be automatically removed during the SAP Business One upgrade process when the corresponding table is modified.

## The SBO\_SP\_TransactionNotification in detail

SAP Business One provides the SBO\_SP\_TransactionNotification stored procedure mechanism for receiving notification of data-driven events. This stored procedure is fired whenever operations are performed on business objects like document or master records in SAP Business One. This applies not only to objects already exposed through DI API, but to any business object in the application – including User-Defined Objects (UDO). By adding your own Transact-SQL code, you can pass parameters to your application and retrieve the corresponding objects from SAP Business One.

The SBO\_SP\_TransactionNotification stored procedure is created automatically when you create a new company in SAP Business One. It is missing from some of the SAP Business One demo databases (for example the US demo database), but you can easily add it manually using SQL Enterprise Manager or SQL Query Analyzer. When it is initially created, it looks like this:

```
CREATE proc SBO_SP_TransactionNotification

@object_type nvarchar(20),          -- SBO Object Type
@transaction_type nchar(1),      -- [A]dd, [U]pdate, [D]elete, [C]ancel, C[L]ose
@num_of_cols_in_key int,
@list_of_key_cols_tab_del nvarchar(255),
@list_of_cols_val_tab_del nvarchar(255)

AS
begin
-- Return values
declare @error int                -- Result (0 for no error)
declare @error_message nvarchar (200) -- Error string to be displayed
select @error = 0
select @error_message = N'Ok'

-----
-- ADD YOUR CODE HERE
-----

-- Select the return values
select @error, @error_message

end
GO
```

There are five out parameters:

Parameter	Example	
<b>@objecttype</b>	2 (Business Partner)	<p>Since version 2005 it returns a string representing the type of object – used to be a numerical value in previous versions.</p> <p>A full list of object types exposed through DI API can be found in the DI API documentation under the section Enumerations. The enumeration is called BoObjectTypes.</p>
<b>@transaction_type</b>	U (update)	<p>It may have values of:</p> <ul style="list-style-type: none"> <li>▪ <b>A</b> (where a record was added)</li> <li>▪ <b>U</b> (where a record was updated)</li> <li>▪ <b>D</b> (where a record was deleted),</li> <li>▪ <b>C</b> (where a document was canceled)</li> <li>▪ <b>L</b> (where a document was closed)</li> </ul>
<b>@num_of_cols_in_key</b>	1	<p>Returns the number of columns or fields in the key to the record.</p> <p>A Business Partner, for example, has a key consisting of a single field (CardCode), so this variable would have a value of "1". A Special Prices object, however, has a key consisting of the CardCode and ItemCode, so for a Special Prices object this variable would have a value of "2".</p>
<b>@list_of_key_cols_tab_del</b>	CardCode	<p>It returns a tab-delimited list of column names (field names) that represent the object key.</p> <p>For a Business Partner, this would be "CardCode". For a Special Prices object, it would be "CardCode ItemCode".</p>
<b>@list_of_cols_val_tab_del</b>	C40001	<p>This returns a tab-delimited list of values required to retrieve the object from Business One.</p> <p>For example, a Business Partner record might have a value of "C40001". A Special Prices object would contain 2 fields separated by a tab character. For example: "V10005 ... A00001"</p>

## Important Note for 2007 version

With the release of SAP Business One 2007, Business One will wait for the SBO\_SP\_TransactionNotification stored procedure to return both parameters @error and @error\_message as return values. Without the return of these values, SAP Business One will consider the transaction as failed.

If you are placing code in the SBO\_SP\_TransactionNotification, please make sure you always go through the "select @error, @error\_message" line at the end, especially in the case where you want the transaction to be considered as successful.

## Adding Code to the Stored Procedure

Now that you are familiar with the values returned by the SBO\_SP\_TransactionNotification stored procedure, let's take a look at a sample code that uses it.

The sample that accompanies this article includes a simple Visual Basic 6 project called "DisplaySBOInfo". You should use REGSVR32.EXE to register the compiled dll that is included. DisplaySBOInfo exposes a single method called DisplaySBOInfo. This method takes the information provided by SBO\_SP\_TransactionNotification as parameters and writes them to a tab-delimited text file called *Log.txt*. The SBO\_SP\_TransactionNotification stored procedure should be modified to include the following Transact-SQL code. Add the following text to the section of the stored procedure that says "Add Your Code Here":

```

DECLARE @object int          --declare the object variable
DECLARE @hresult int        --declare the hresult variable
DECLARE @retval float       --declare the return variable

EXEC @hresult = sp_OACreate 'DisplaySBOInfo.Class1', @object OUT
EXEC @hresult = sp_OAMethod @object, DisplaySBOInfo, @retval OUT, @object_type,
@transaction_type, @num_of_cols_in_key, @list_of_key_cols_tab_del,
@list_of_cols_val_tab_del

IF @hresult <> 0
BEGIN
EXEC sp_OAGetErrorInfo @object
RETURN
END

```

This creates an instance of `DisplaySBOInfo.Class1` and calls the `DisplaySBOInfo` method, passing parameters to it that will be written to the text log.

After you have added your code to the stored procedure, open up SAP Business One and make a few changes. For example, add a Business Partner, edit a Quotation, and then delete the Business Partner. When you're done, open *Log.txt* and take a look at the data. In a real-life scenario, you would use the information provided by SBO\_SP\_TransactionNotification with the DI-API to retrieve more information about the record that was inserted, deleted, or changed.

When you are done with this sample exercise, go back to the stored procedure and delete or comment out the code that was added. This will insure that the sample does not continue to run.

If you would like, you can [download the sample code](#) that was described in this article.

You can also download [SAP Business One DI Event Service](#) sample implementing a DI Event Service that runs on top of the existing SAP Business One SDK interfaces. This service provides notifications on events related to SAP Business One DI API objects through a listener-based interface.

## Related Contents

For more information, visit the [Business One homepage](#).

## Copyright

© Copyright 2009 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors. Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.