

Tags involved in transaction. Only one tag is actively monitored. The other tags are Read when transaction is activated. Tag "TransReq" triggers the transaction. Tag "TransCode" contains the result of the transaction. Also note that a stored procedure is used, "ospInsertData".

The screenshot displays the Ignition Designer interface for a transaction named "SPTrans".

Project Browser: Shows the project structure including Configuration, Transaction Groups, and SPTrans.

SQLTags Browser: Lists various tags with their values and data types:

Tag	Value	Data Type
Data1	2,013	Int4
Data2	47	Int4
Data3	21	Int4
Data4	53,120	Int4
DBConection	0	Int4
TransCode	0	Int4
TransReq	<input type="checkbox"/>	Boolean
DBAvailable	1	Int4

SPTrans Running: Shows the configuration for the transaction's data sources and outputs:

Item Name	Value	Target Name	Output	Data Type	Properties
BatchTags>Data1	State	Data1	None	Int4	
BatchTags>Data2	State	Data2	None	Int4	
BatchTags>Data3	State	Data3	None	Int4	
BatchTags>Data4	State	Data4	None	Int4	
BatchTags/TransCode	State	Read-only	None	Int4	
BatchTags/TransReq	false	Read-only	None	Boolean	
Gateway/CurrentDateTime	State	TStamp	None	DateTime	

Run-Always Expression Items (ignore trigger): (0)

Triggered Expression Items (0)

Execution Scheduling: Set to 250 milliseconds. Data source: Nucor_IgnitionDB. Procedure name: ospInsertData.

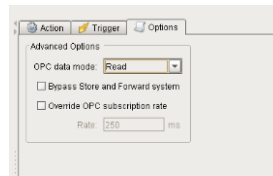
Status: Running. Last execution: Wed Nov 20 08:47:31 CST 2013. Total executions: 0. Failed executions: 0. Next execution: Wed Nov 20 08:47:31 CST 2013. OPGTag writes: 0. Last duration: 0.0 second(s). DB writes: 0. Avg duration: 0.0 second(s). OPGTag write failures: 0.

Trigger tab allows you to set the trigger options. Also have set the only tags to evaluate to the trigger tag, "TransReq". This also allows you to set the values for a Success or Failure.

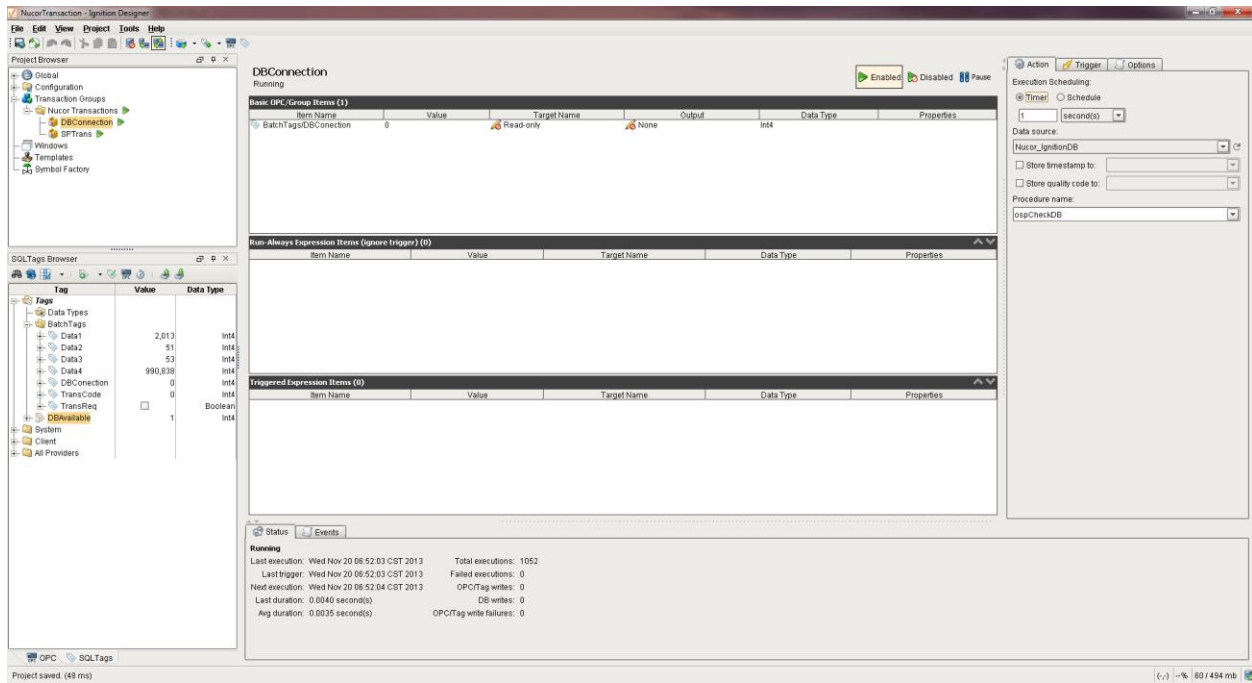
The screenshot shows the 'Trigger' tab of an Ignition Transaction Group configuration window. The window has three tabs: 'Action', 'Trigger', and 'Options'. The 'Trigger' tab is active. The configuration includes the following options:

- Only evaluate when values have changed
- Tags to watch for change: Custom... (with a 'Select tags' button)
- Execute this group on a trigger
- Trigger on item: BatchTags/TransReq (dropdown menu)
- Only execute once while trigger is active
- Reset trigger after execution
- Prevent trigger caused by group start
- Trigger conditions:
 - is > 0
 - is = 0
 - is active: > 0 (dropdown) 0 (text box)
 - non-active: <= 0 (dropdown) 1 (text box)
 - Active on value change
- Write handshake on success
 - Set: BatchTags/TransCode (dropdown)
 - To value: 100 (text box)
- Write handshake on failure
 - Set: BatchTags/TransCode (dropdown)
 - To value: 500 (text box)

Option tab allows you to set the OPC mode to Read, which causes it to only actively read the “TransReq” and the other items are a forced read at the transaction time to make sure data is correct. Note that “Bypass Store and Forward system” is not selected. This allows unsuccessful transactions to be cached until the database is online again. The transactions will be executed in the order they entered the cache (first in first out).



This transaction is to verify the function of the database. It executes every one second and calls a stored procedure that only contains a single select statement, "Select 1". The PLC tag is set to Read-only, since we are writing to that tag, based on a successful database connection, set in the trigger tab.

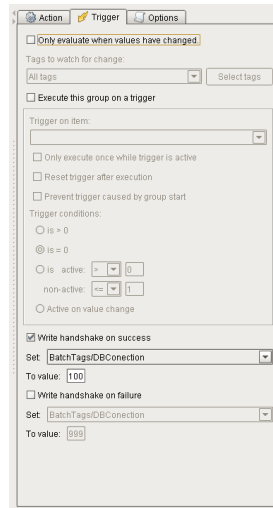


The screenshot displays the Ignition Designer interface for configuring a transaction group named 'DBConnection'. The main window is titled 'DBConnection Running' and shows the following configuration details:

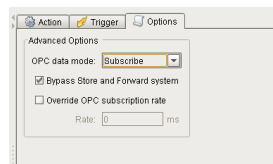
- Basic OPC_Group Items (1):** A table with one item:

Item Name	Value	Target Name	Output	Data Type	Properties
BatchTags/DBConnection	0	Read-only	None	Int4	
- Run-Always Expression Items (Ignore Triggers) (0):** An empty table.
- Triggered Expression Items (0):** An empty table.
- Execution Scheduling:**
 - Execution mode: Timer, Schedule
 - Interval: 1 (seconds)
 - Data source: Huxor_IgnitionDB
 - Store timestamp to:
 - Store quality code to:
 - Procedure name: ospCheckDB
- Tag Browser:** Shows a tree view of tags. The 'DBConnection' tag is highlighted with a value of 0 and data type of Int4.
- Status/Events:**
 - Running
 - Last execution: Wed Nov 20 08:52:03 CST 2013
 - Total executions: 1052
 - Last trigger: Wed Nov 20 08:52:03 CST 2013
 - Failed executions: 0
 - Next execution: Wed Nov 20 08:52:04 CST 2013
 - OPCTag writes: 0
 - Last duration: 0.0040 second(s)
 - DB writes: 0
 - Avg duration: 0.0035 second(s)
 - OPCTag write failures: 0

The only option we set is the “Write handshake on success” and enter an appropriate value. Have basic PLC code to reset and use value as a heartbeat.



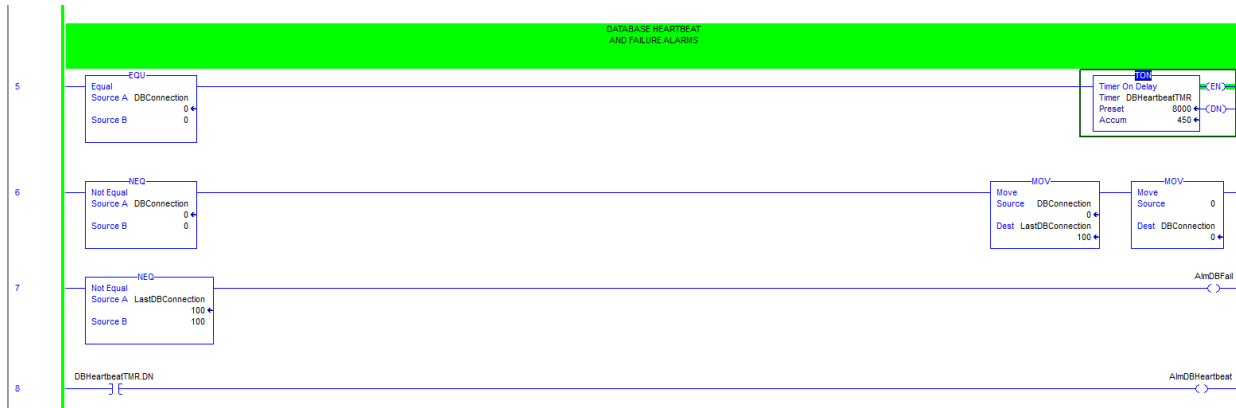
On this one the OPC data mode is set to subscribe and the “Bypass Store and Forward system” is selected. Since this is only validation of database connection the transaction does not need to be cached if the database is offline.



PLC transaction trigger and reset.



PLC database heartbeat and alarm bit.



ospInsertData stored procedure.

```
-- Author:      Darren Ash
-- Create date: 11/19/2013
-- Description: Used to insert data into a table from Ignition.
--             Could be used to insert data into multiple tables from one
--             stored procedure.
-----
ALTER PROCEDURE [dbo].[ospInsertData]
-- Add the parameters for the stored procedure here
@Data1 int,
@Data2 int,
@Data3 int,
@Data4 int,
@TStamp DateTime
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for procedure here
INSERT INTO tblTrans(Data1, Data2, Data3, Data4, InsertTime) VALUES (@Data1, @Data2, @Data3, @Data4, @TStamp)
END
```

ospCheckDB stored procedure.

```
-- -----
-- Author:      Darren Ash
-- Create date: 11/20/2013
-- Description: Ignition executes a "Select 1" to verify the database is connected and online
-- -----
ALTER PROCEDURE [dbo].[ospCheckDB]
-- Add the parameters for the stored procedure here
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for procedure here
SELECT 1
END
```