

Tech Note 300

Automatically Reconnecting to an Outside Data Source in Industrial Application Server

All Tech Notes and KBCD documents and software are provided "as is" without warranty of any kind. See the **Terms of Use** for more information.

Topic#:001107

Created: January, 2003

Updated: March, 2004

Problem/Issue

Industrial Application Server has no function to automatically re-connect to disconnected I/O Servers.

Solution 1

Automatic re-connection to disconnected I/O Servers can be accomplished on a DIObject-by-DIObject basis or by simply creating a copy of the base DIObject, in this case the **\$DDESuiteLinkClient** object, and creating a reconnect script in it.

When you create new DIObjects from this template you will automatically inherit the reconnect script.

Simply use the following script in each instance of a DIObject:

```
Me.Connectionstatus <> "Connected";
```

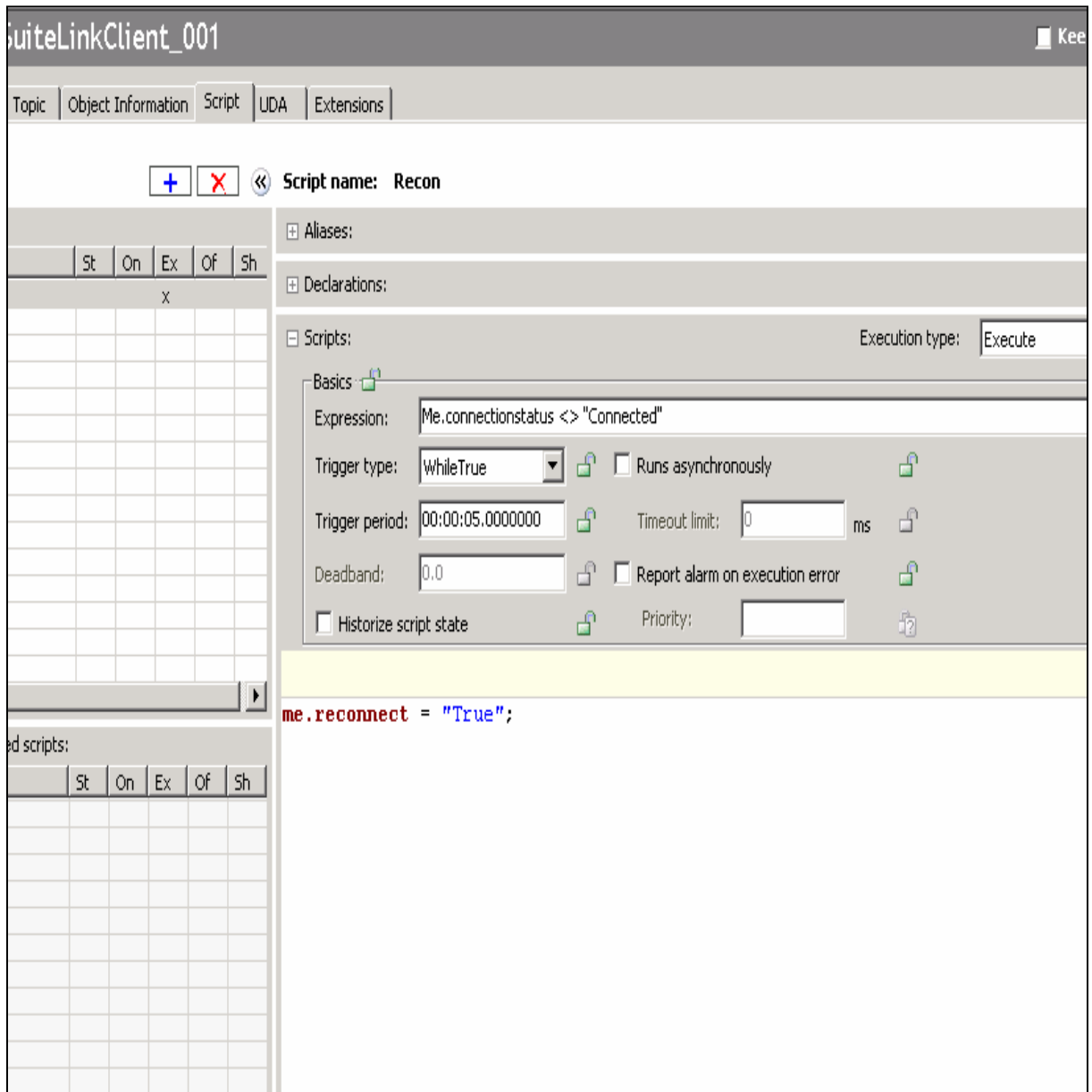


FIGURE 1: DI OBJECT SCRIPT

The default Expression is:

```
<DIObjectName>.Connectionstatus <> "Connected"
```

In the above figure:

Expression:

```
Me.Connectionstatus <> "Connected";
```

In this example the word **me** is used so that you can create as many instances as necessary and not have to change the script. The keyword me will always reference the object itself.

Trigger Type: While True

Trigger Period: 00:00:05.000000 (5 Seconds)

Script body:

```
<DIObjectName>.Reconnect = 1;
```

In this example: **me.Reconnect = 1;**

This script will run every 5 seconds and try to force the DIObject to reconnect to the IO Server.

Solution 2

To minimize the amount of network traffic generated by constant reconnect attempts to IO Sources that may be offline, simply use the solution below.

Create 6 UDAs in the parent DI Object so that they are inherited. The UDA's created should be

UDA Name	Type	Init Value	Extension
AttemptCount	Integer	3	*
ConnAttempt	Integer	0	*
RetryCount	Integer	30	*
TotalRetries	Integer	0	History Extended
ConnectionCount	Integer	0	History Extended
DisconnectionCount	Integer	0	History Extended

Modify the script in Solution 1 to read as follows:

Trigger Type: While True

Trigger Period: 00:00:05.000000 (5 Seconds)

Script body:

```
Me.ConnAttempt = Me.ConnAttempt + 1;
If Me.ConnAttempt < Me.AttemptCount then
Me.Reconnect =1;
Me.TotalRetries = Me.TotalRetries +1;
Endif;

If Me.ConnAttempt > Me.RetryCount then
Me.ConnAttempt =0;
Endif;
```

This script will try and connect to the IO Source 3 times and then wait for 140 seconds and then retry the reconnect another 3 times before waiting another 140 seconds. The RetryCount can be modified to allow a longer or greater wait period depending on the trigger period and your specific reconnection needs.

Another script can also be written to help track how many connections and disconnections have taken place:

Expression: `Me.Connectionstatus <> "Connected";`

In this example the word **me** is used so that you can create as many instances as necessary and not have to change the script. The keyword **me** will always reference the object itself.

Trigger Type: On True

Script body:

```
Me.ConnectionCount = Me.ConnectionCount + 1;
Me.ConnAttempt = 0;
```

Trigger Type: On False

Script body:

```
Me.DisconnectionCount = Me.DisconnectionCount + 1;
```

The three UDAs that have been history extended can be logged to InSQL and trended to see

where any potential problems may occur. This would be useful for tuning the system and maximizing network bandwidth.

R. Ekstein and B. Hunter

Tech Notes are published occasionally by Wonderware Technical Support. Publisher: Invensys Systems, Inc., 26561 Rancho Parkway South, Lake Forest, CA 92630. There is also technical information on our software products at **Wonderware Technical Support**

For technical support questions, send an e-mail to support@wonderware.com.



back to top

©2015 Invensys Systems, Inc. All rights reserved. No part of the material protected by this copyright may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, broadcasting, or by any information storage and retrieval system, without permission in writing from Invensys Systems, Inc. **Terms of Use.**