

Tech Note 933

Scripting the InBatch Phase Object

All Tech Notes, Tech Alerts and KBCD documents and software are provided "as is" without warranty of any kind. See the [Terms of Use](#) for more information.

Topic#: 002750

Created: February 2013

Introduction

Typically the InBatch Phase object is tied to I/O in order to implement phase logic, but it can also be used to implement a script-based phase. Examples where this can be useful are performing actions in an external system and/or getting data from an external system for use within an executing batch. In this scenario, the phase object will execute a script when commanded to run by the InBatch Batch Manager. Data can be passed to and back from the script using phase parameters.

In order to create a script-based phase, you must create a series of Application Object scripts in order to perform the handshaking between the InBatch Batch Manager and the InBatch Phase object. Typically only the minimum phase control/status tags (Ready, Start, Done, Reset) are needed to implement a script phase, so only a subset of the scripts listed below are needed. The full set of control/status tags are only needed in the rare case where the script action being performed is very long running and the ability to hold and/or abort the script is desired.

[Download the InBatchPhaseScript.aaPKG file \(zipped\)](#).

Application Versions

- Wonderware InBatch 2012 R2 (v. 10)

OnScan

This script initializes the state of phase status bits when the InBatch Phase object is first put on scan. The **Me.Phase.Aborted**, **Me.Phase.Held**, **Me.Phase.Interlocked** and **Me.Phase.Run** tags are optional.

The **Me.Phase.Interlocked** tag can be set to make the phase unavailable for execution by InBatch.

Script: **OnScan**

Execution type: **OnScan**

```
Me.Phase.Aborted = 0;  
Me.Phase.Done = 0;  
Me.Phase.Held = 0;  
Me.Phase.Interlocked = 0;  
Me.Phase.Run = 0;  
Me.Phase.Ready = 1;
```

Start

Scripting the InBatch Phase Object

This is where the real work is performed. It will be called when the InBatch BatchMngr commands the phase to start. Data can be passed from the InBatch Batch Manager to this script and back using phase parameters.

It is recommended that this script run asynchronously, especially if making an external call.

Script: **Start**

Execution type: **Execute**

Expression: **Me.Phase.Start**

Trigger type: **OnTrue**

Runs asynchronously: **Checked**

```
if Me.Phase.Ready == 1 then
  Me.Phase.Ready = 0;
  Me.Phase.Run = 1;

  ' Example code below, replace with your own code.

  dim connection as System.Data.SqlClient.SqlConnection;
  dim command as System.Data.SqlClient.SqlCommand;
  dim reader as System.Data.SqlClient.SqlDataReader;

  connection = new System.Data.SqlClient.SqlConnection("server=(local);uid=sa;database=lookup");
  connection.Open();
  command = new System.Data.SqlClient.SqlCommand("select material from materials where id = " + Me.Id.Target, connection);
  reader = command.ExecuteReader();
  reader.Read();
  Me.Result.Actual = reader("material");
  reader.Close();
  connection.Close();

  ' End of example code

  Me.Phase.Run = 0;
  Me.Phase.Done = 1;
endif;

Me.Phase.Start = 0;
```

Reset

This script resets the state of the phase when commanded by the InBatch Batch Manager. The Me.Phase.Aborted tag is optional.

Script: **Reset**

Execution type: **Execute**

Expression: **Me.Phase.Reset**

Trigger type: **OnTrue**

```
if Me.Phase.Done == 1 OR Me.Phase.Aborted == 1 then
  Me.Phase.Done = 0;
  Me.Phase.Aborted = 0;
  Me.Phase.Ready = 1;
endif;

Me.Phase.Reset = 0;
```

Hold

This script is optional, and is only needed if the Hold control tag is configured for the phase in InBatch.

Script: **Hold**

Execution type: **Execute**

Expression: **Me.Phase.Hold**

Trigger type: **OnTrue**

```
if Me.Phase.Run == 1 then
  ' Add code here to pause your script execution

  Me.Phase.Run = 0;
  Me.Phase.Held = 1;
endif;

Me.Phase.Hold = 0;
```

Restart

This script is optional, and is only needed if the Restart control tag is configured for the phase in InBatch.

Script: **Restart**

Execution type: **Execute**

Expression: **Me.Phase.Restart**

Trigger type: **OnTrue**

```
if Me.Phase.Held == 1 then
  ' Add code here to restart your script execution

  Me.Phase.Held = 0;
  Me.Phase.Run = 1;
endif;

Me.Phase.Restart = 0;
```

Abort

This script is optional, and is only needed if the Abort control tag is configured for the phase in InBatch.

Script: **Abort**

Execution type: **Execute**

Expression: **Me.Phase.Abort**

Trigger type: **OnTrue**

```
if Me.Phase.Held == 1 then
  ' Add code here to abort your script execution

  Me.Phase.Held = 0;
  Me.Phase.Aborted = 1;
endif;

Me.Phase.Abort = 0;
```

E. Grove

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 wwsupport@invensys.com.

 [Back to top](#)

©2013 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](#).