

[Tech Note 982](#)

Upgrading InBatch History Server from 8.1 SP1 to 2012 R2

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#: 002808

Created: October 2013

Introduction

This *Tech Note* outlines the procedure to upgrade InBatch History Server from 8.1 (including SP1) to InBatch History Server 10. By design, you upgrade from 8.x to 9.x, then to 10.

However, you can use this Tech Note to upgrade from IB 8.1 or IB 8.1 SP directly to IB 2012 R2.

Application Versions

- InBatch History Server from 8.1 or IB 8.1 SP1
- InBatch 2012 R2 (IB 10.0)

Modify the BatchInfo Table

Configurator.exe checks the **BatchInfo** table.

- For 8.1 SP1, if the **BatchInfo** table exists, delete the table in both the **BatchHistory** and **BatchArchive** databases.

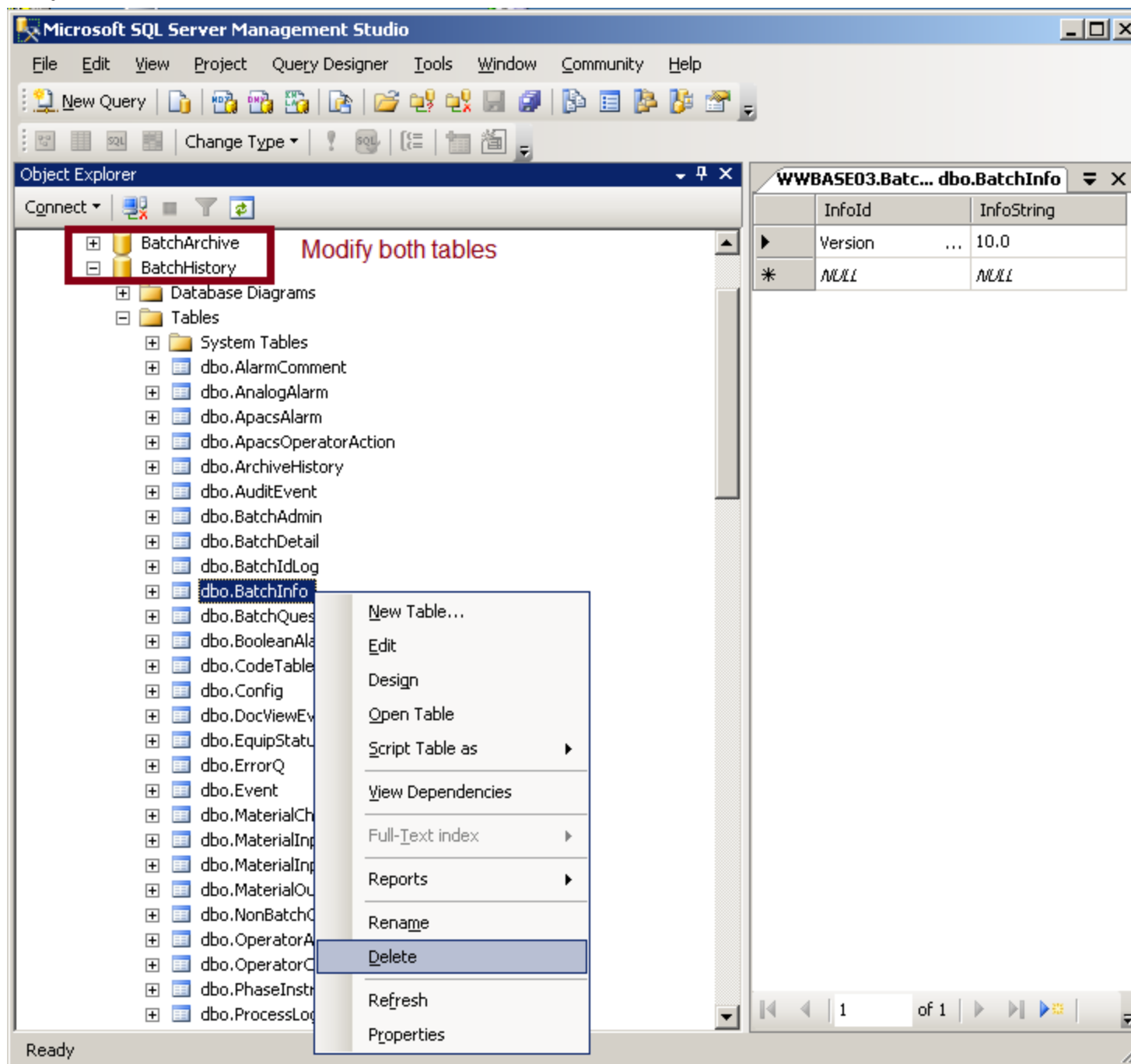


FIGURE 1: DELETE THE BATCHINFO TABLE IN BOTH DATABASES

Replace the AlterBatchServerNameFieldAdd Script

InBatch uses the identical **AlterBatchServerNameFieldAdd.sql** script in two places. To locate it, search **C:\Program Files\Wonderware\InBatch\BatchHistoryServer**.

The modified script contains:

- Additional exist-checks in order for scripts to run again even if some parts already succeeded.
- Additional 'go' statements to breakup larger transactional sections into smaller parts.

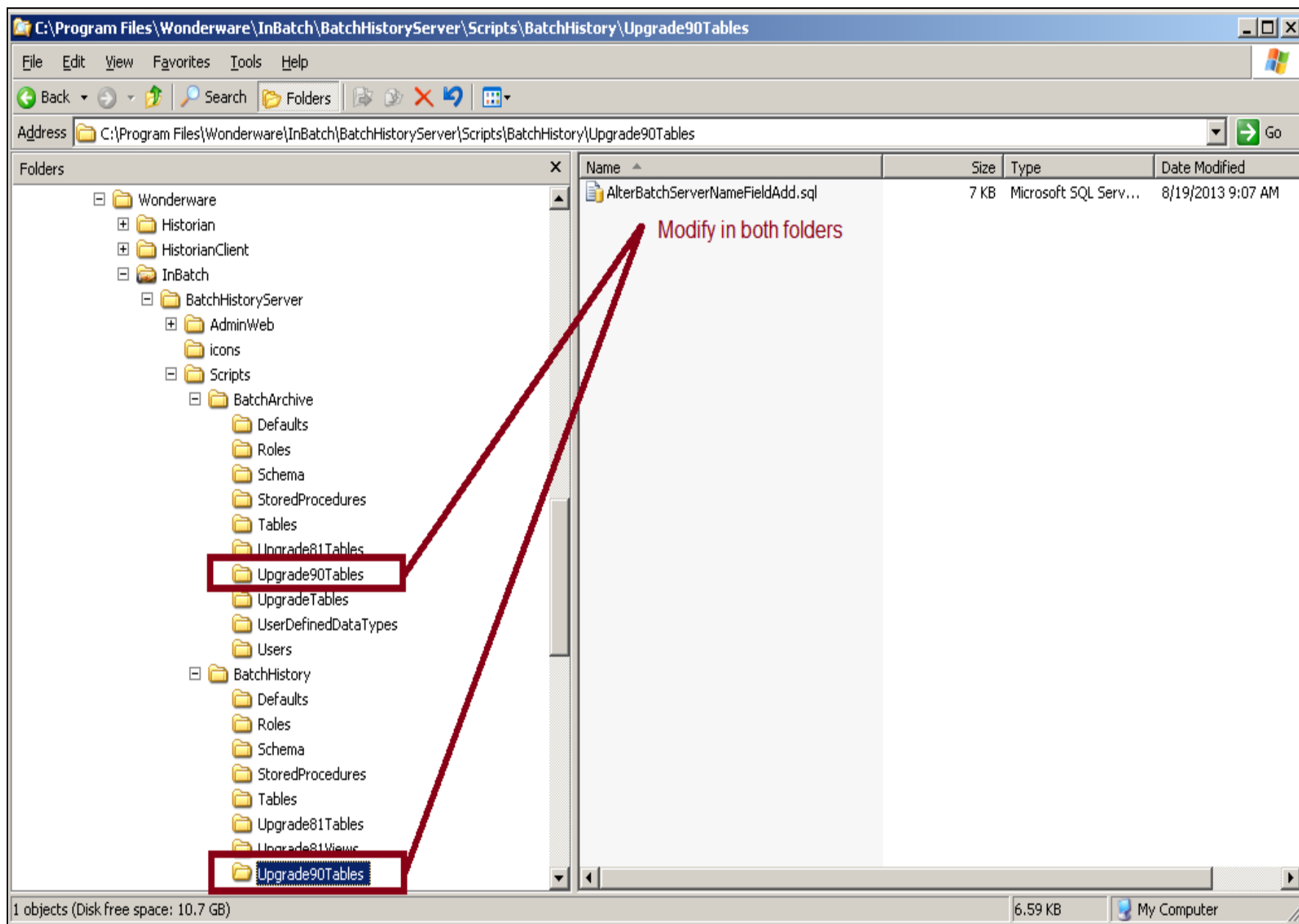


FIGURE 2: MODIFY THE UPGRADE SCRIPTS

Execute Script in SQL Management Studio

If possible, execute the following script in SQL Management Studio first, before **Configurator.exe** executes it. This allows us to follow the progress and to ensure that the long-running scripts complete successfully.

Modified Long-Running Script (AlterBatchServerNameFieldAdd.sql)

```

/***** START : Use Story 1261 : Support for Recipe Name length 128 characters *****/
/***** Object: UserDefinedDataType [dbo].[Recipe_Name] Size changed to 128 *****/

CREATE TYPE [dbo].[Recipe_Name2]
FROM [varchar](128) NOT NULL
GO

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[BatchIdLog]'))
ALTER TABLE BatchIdLog Alter Column Recipe_Name Recipe_Name2
GO

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[BatchIdLog_Arch]'))
EXEC sp_refreshview BatchIdLog_Arch
GO

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[IBBatchJournal]'))
EXEC sp_refreshview IBBatchJournal
GO

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[IBBatchListing]'))
EXEC sp_refreshview IBBatchListing
GO

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[USP_BatchIDLogAdd]'))
DROP PROCEDURE USP_BatchIDLogAdd
GO

DROP TYPE [dbo].[Recipe_Name]
GO

CREATE TYPE [dbo].[Recipe_Name]
FROM [varchar](128) NOT NULL
GO

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[BatchIdLog]'))
ALTER TABLE BatchIdLog
ALTER COLUMN Recipe_Name Recipe_Name
GO

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[BatchIdLog_Arch]'))
EXEC sp_refreshview BatchIdLog_Arch
GO

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[IBBatchJournal]'))
EXEC sp_refreshview IBBatchJournal
GO

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[IBBatchListing]'))
EXEC sp_refreshview IBBatchListing
GO

DROP TYPE [dbo].[Recipe_Name2]
GO

/***** END : Use Story 1261 : Support for Recipe Name length 128 characters *****/
/***** Object: UserDefinedDataType [dbo].[BatchServer_Name] Creation *****/
/* Change made - added check for already exists otherwise script aborts here */

IF TYPE_ID(N'BatchServer_Name') IS NULL
CREATE TYPE [dbo].[BatchServer_Name]
FROM [varchar](20) NOT NULL;
GO

SET ANSI NULLS OFF

```

Upgrading InBatch History Server from 8.1 SP1 to 2012 R2

```

GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

/***** Alter tables with Batch_Server_Name field *****/
/* Change made - added check for already exists otherwise script aborts here */
/* Change made - added 'go' after each table for smaller transactions */

IF COL_LENGTH('[AlarmComment]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[AlarmComment]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[AnalogAlarm]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[AnalogAlarm]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[AuditEvent]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[AuditEvent]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[BatchDetail]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[BatchDetail]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[BatchIdLog]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[BatchIdLog]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[BatchQuestion]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[BatchQuestion]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[BooleanAlarm]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[BooleanAlarm]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[DocViewEvent]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[DocViewEvent]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[EquipStatus]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[EquipStatus]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[Event]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[Event]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[MaterialChar]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[MaterialChar]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[MaterialInput]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[MaterialInput]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[MaterialInputChange]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[MaterialInputChange]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[MaterialOutput]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[MaterialOutput]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[NonBatchOperatorActions]', 'Batch_Server_Name') IS NULL

```

Upgrading InBatch History Server from 8.1 SP1 to 2012 R2

```
ALTER TABLE [dbo].[NonBatchOperatorActions]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[OperatorActions]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[OperatorActions]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[OperatorComment]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[OperatorComment]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[PhaseInstruction]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[PhaseInstruction]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[ProcessLog]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[ProcessLog]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[ProcessVar]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[ProcessVar]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[ProcessVarChange]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[ProcessVarChange]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[SequenceBlock]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[SequenceBlock]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[Transition]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[Transition]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[TransitionExpression]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[TransitionExpression]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

IF COL_LENGTH('[UserProfile]', 'Batch_Server_Name') IS NULL
ALTER TABLE [dbo].[UserProfile]
ADD [Batch_Server_Name] [dbo].[BatchServer_Name] NOT NULL DEFAULT (N' ');
GO

SET ANSI_PADDING OFF
```

1. Execute this script against both the **BatchHistory** and **BatchArchive** databases.

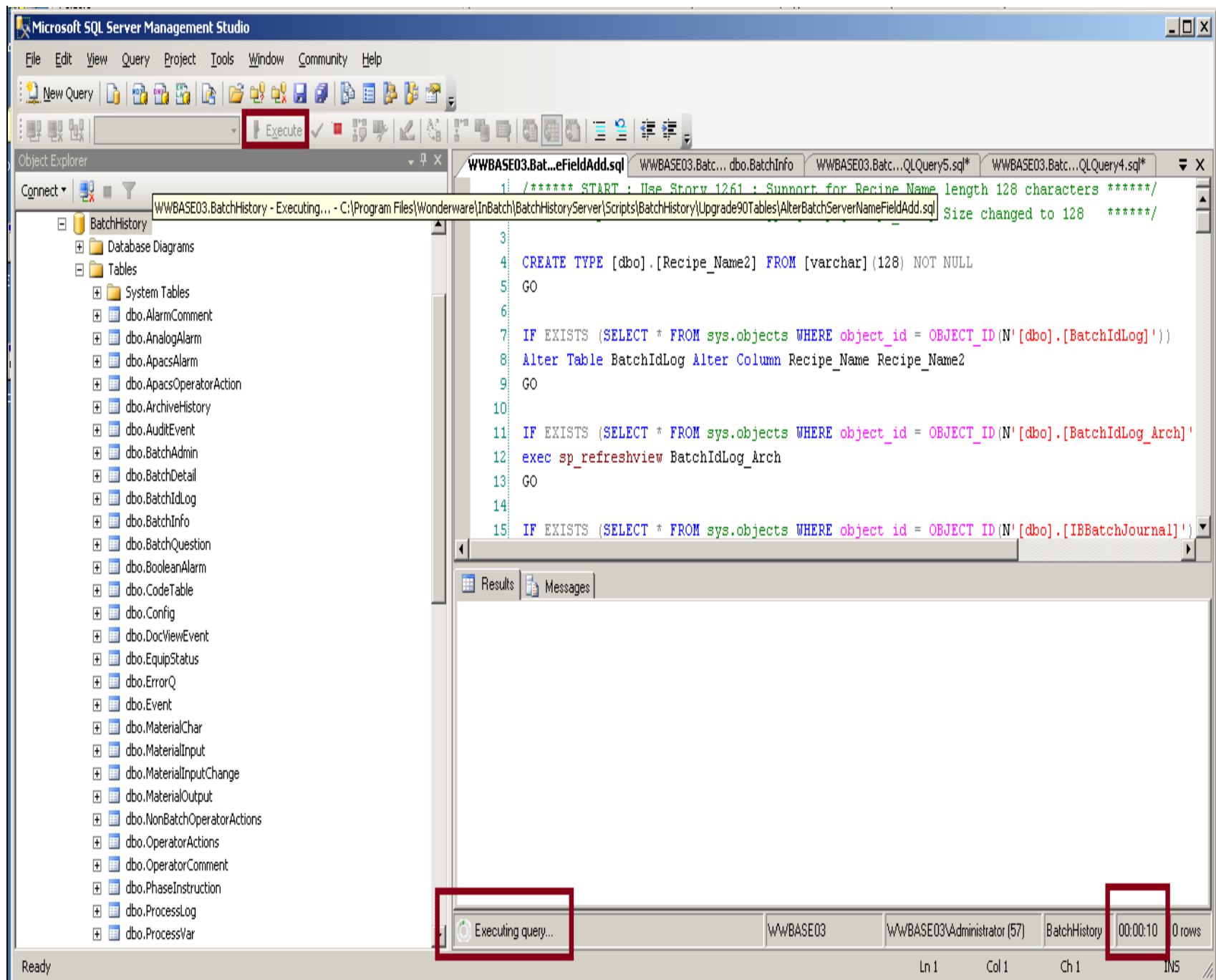


FIGURE 3: MONITOR SCRIPT EXECUTION

2. Execute **Configurator.exe** again.

3. After you reset the **BatchInfo** table, and after the modified scripts are place and possibly the long-running scripts already executed, re-run **Configuror.exe**.
4. Select the InBatch History Server and press **Configure** button.

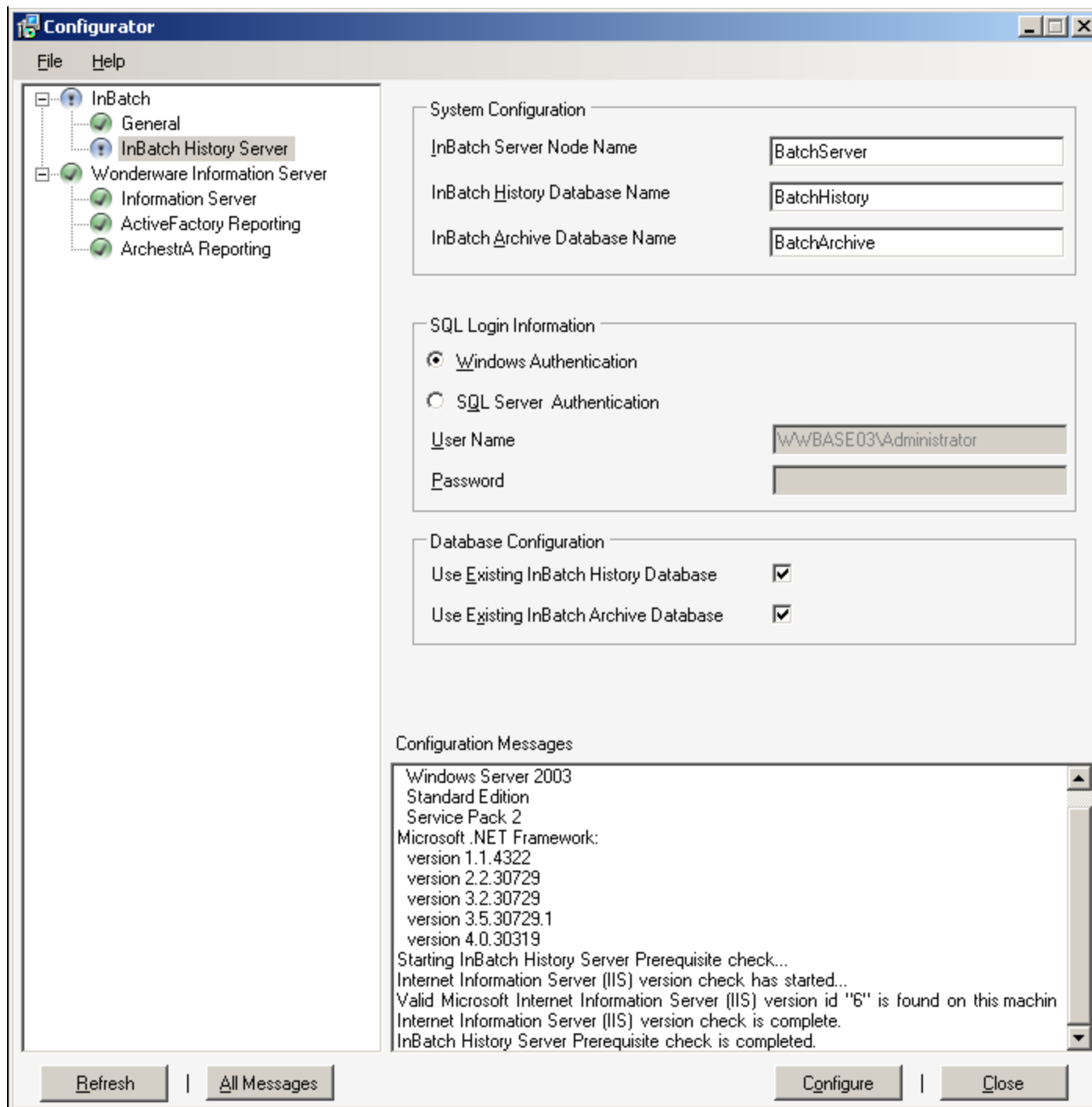


FIGURE 4: INBATCH CONFIGURATOR

Possible Issues with Configurator

You might encounter the following scenarios while updating:

Issue: Failure on upgrading Stored Procedures

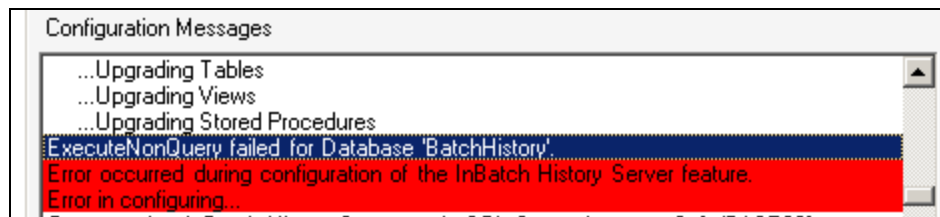


FIGURE 5: QUERY FAILED FOR DATABASE 'BATCHHISTORY'

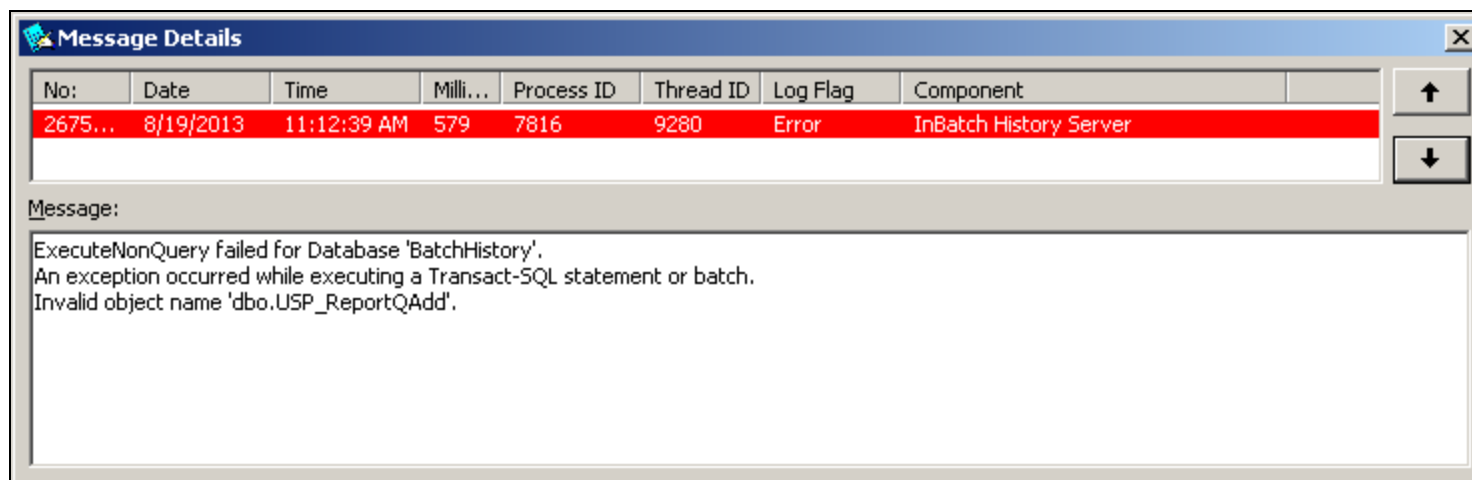


FIGURE 6: MESSAGE DETAILS

Resolution: Manually execute the Stored Procedure creation prior to upgrades being executed by Configurator.exe

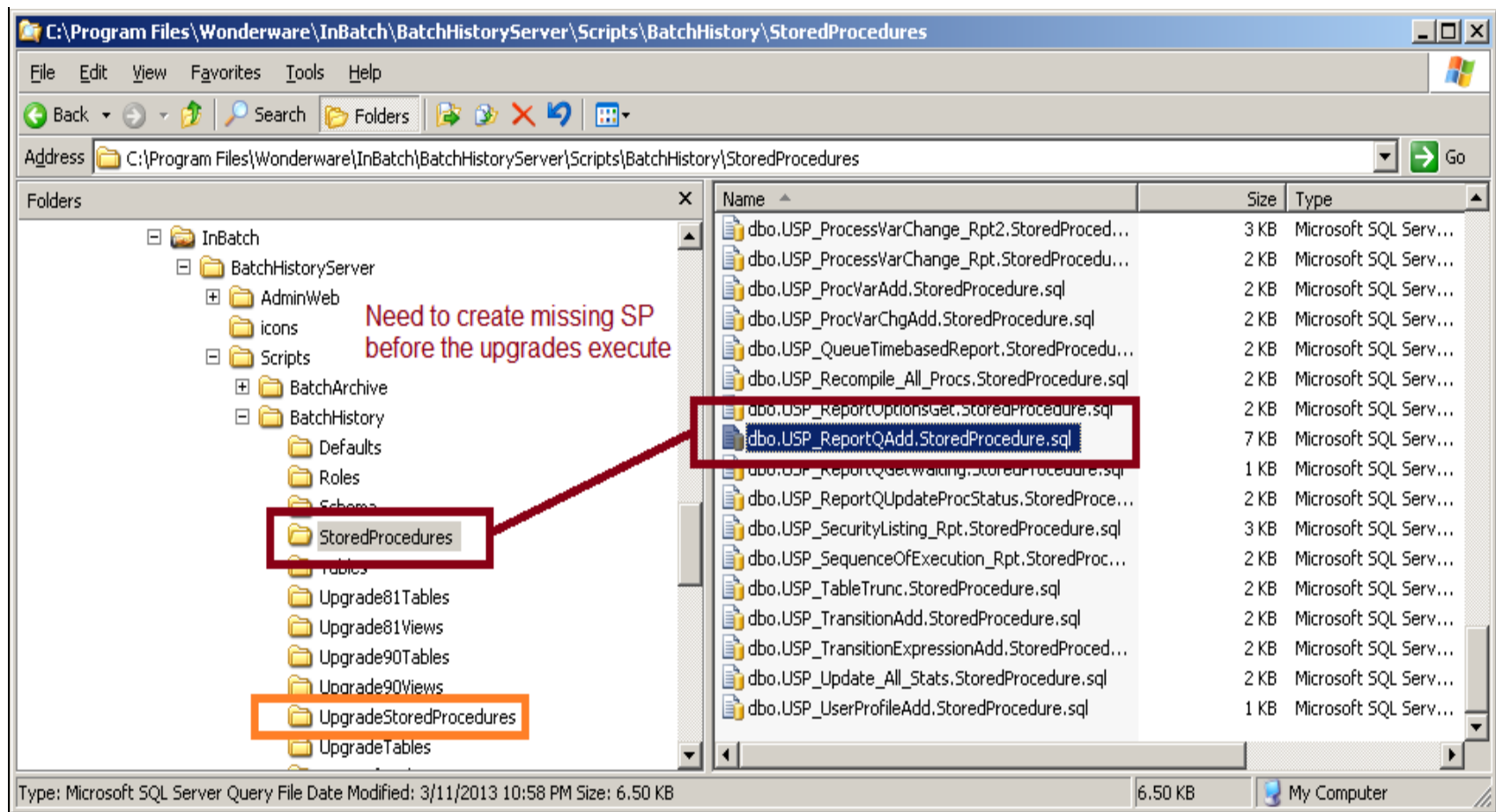


FIGURE 7: MANUALLY EXECUTE USP BEFORE UPGRADING

Issue: Failure on upgrading Views

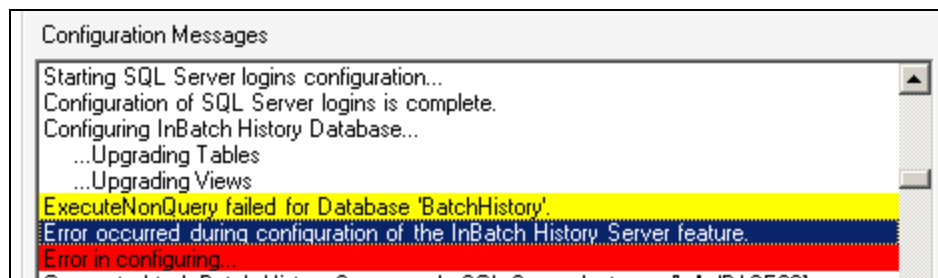


FIGURE 8: FAIL ON CONFIGURING VIEWS

Resolution: Delete all Views (one at a time if necessary).

Configurator will re-create Views and then continue with rest of configuration steps.

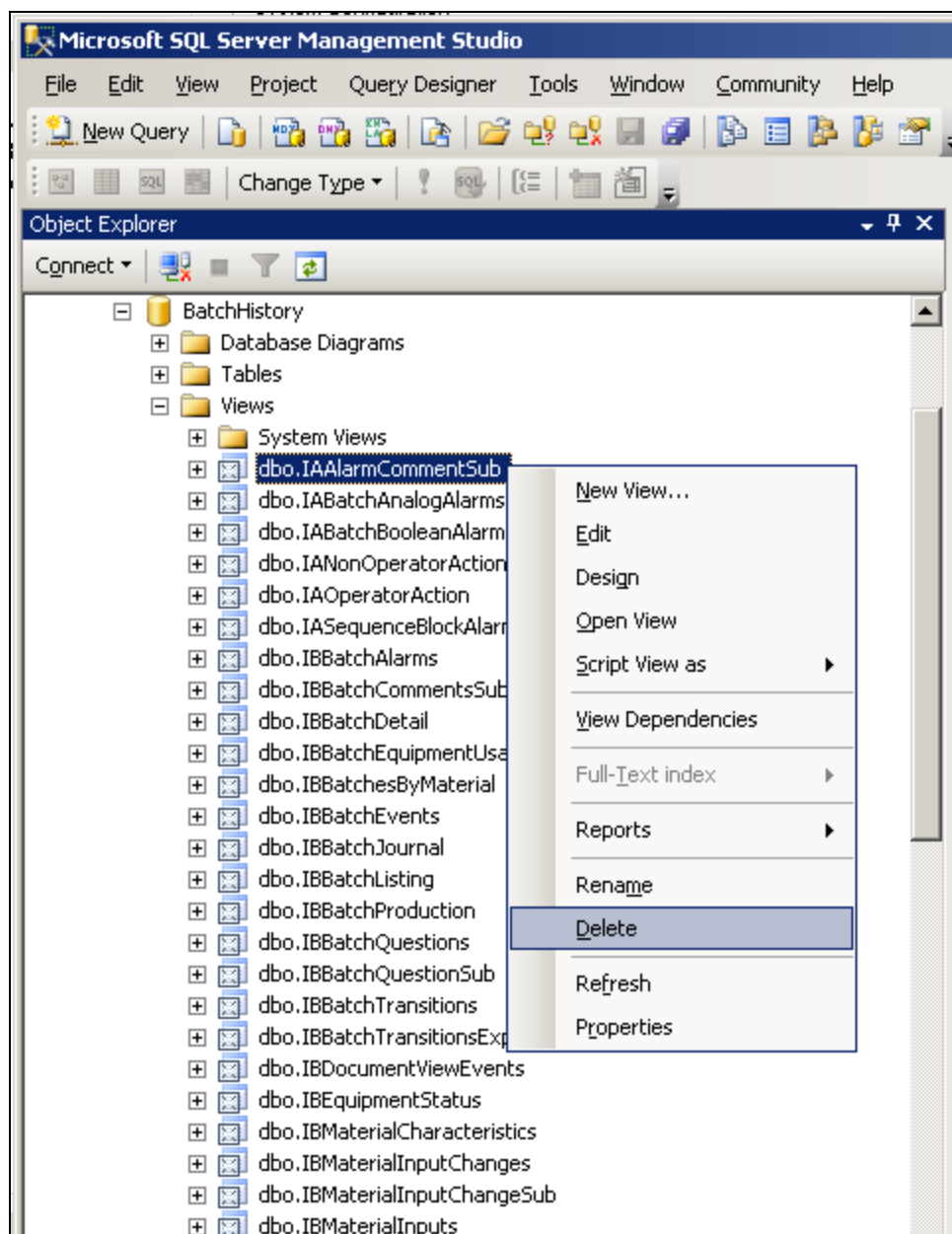


FIGURE 9: DELETE INBATCH VIEWS

Issue: Error creating History Database Administration Start menu item

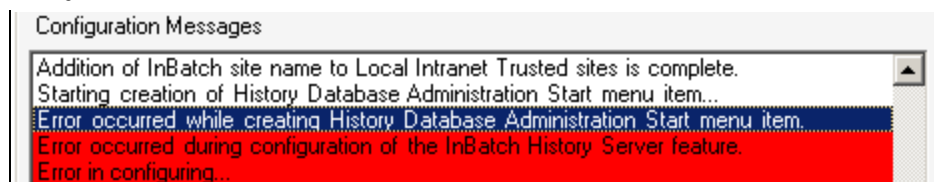


FIGURE 10: ERROR CONFIGURING ADMIN START ITEM

Resolution: Create a missing registry setting and re-run Configurator.exe

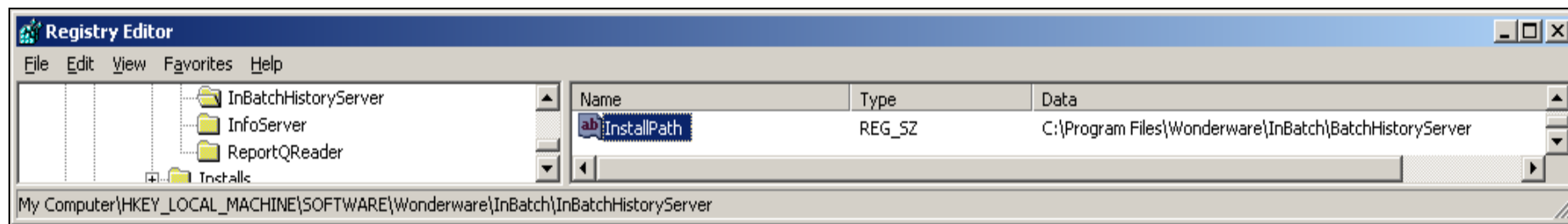


FIGURE 11: CREATE INBATCHHISTORYSERVER INSTALLPATH REGISTRY ITEM

Upgrading config_A

When upgrading from IB 8.1 to 2012 or 2012R2 you must also complete the following steps.

1. Backup and save the **config_A** file.
2. Uninstall InBatch 8.1 and clean up all the directories for InBatch. The default directory location is **C:\Program Files\FactorySuite\InBatch**.

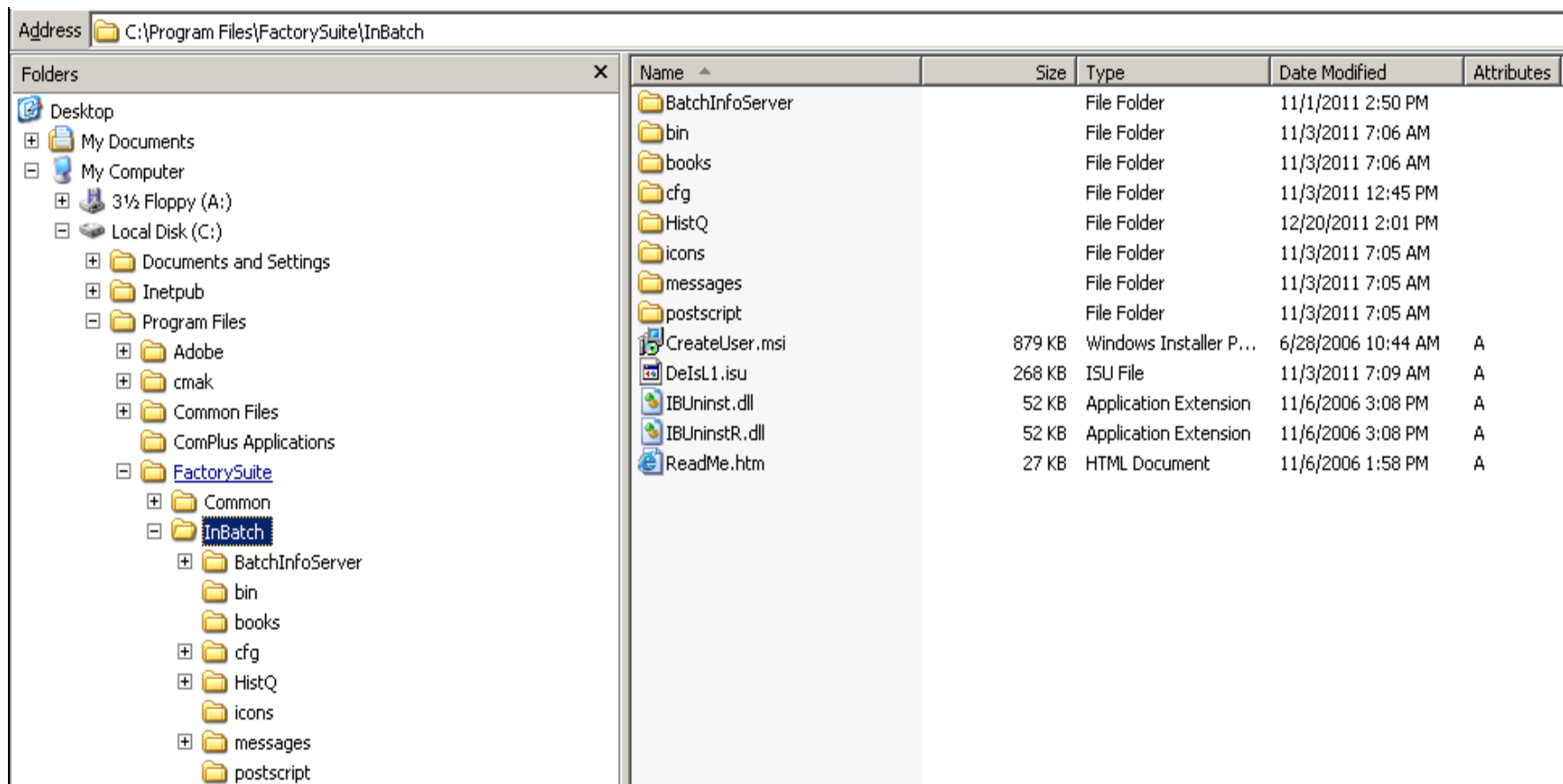


FIGURE 12: INBATCH DIRECTORIES

- Now you can upgrade the OS if needed or you can use the supported OS that is installed.
- Next, install the correct version of InBatch (2012 or 2012 R2).
- After the install is complete, you need to convert the **config_A** file.
- In the new system you will need to replace the existing **config_A** with the **InBatch 8.1** config_A, and then run the **cvtmaster**.

In the new 2012or 2012 R2 system the config_A is located **C:\Program Files\Wonderware\InBatch\Cfg\Config_A**.

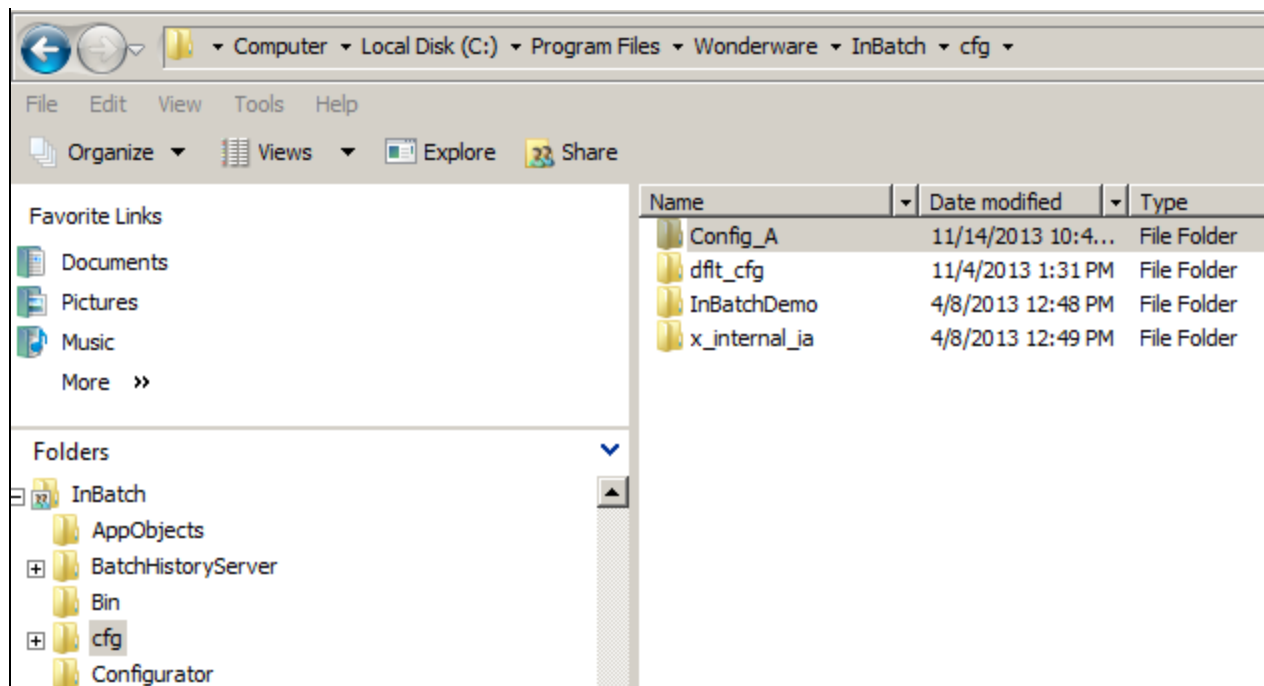


FIGURE 13: NEW CONFIG_A FILE LOCATION

7. Run the **cvtmaster** and convert the config_A to the current version of InBatch.

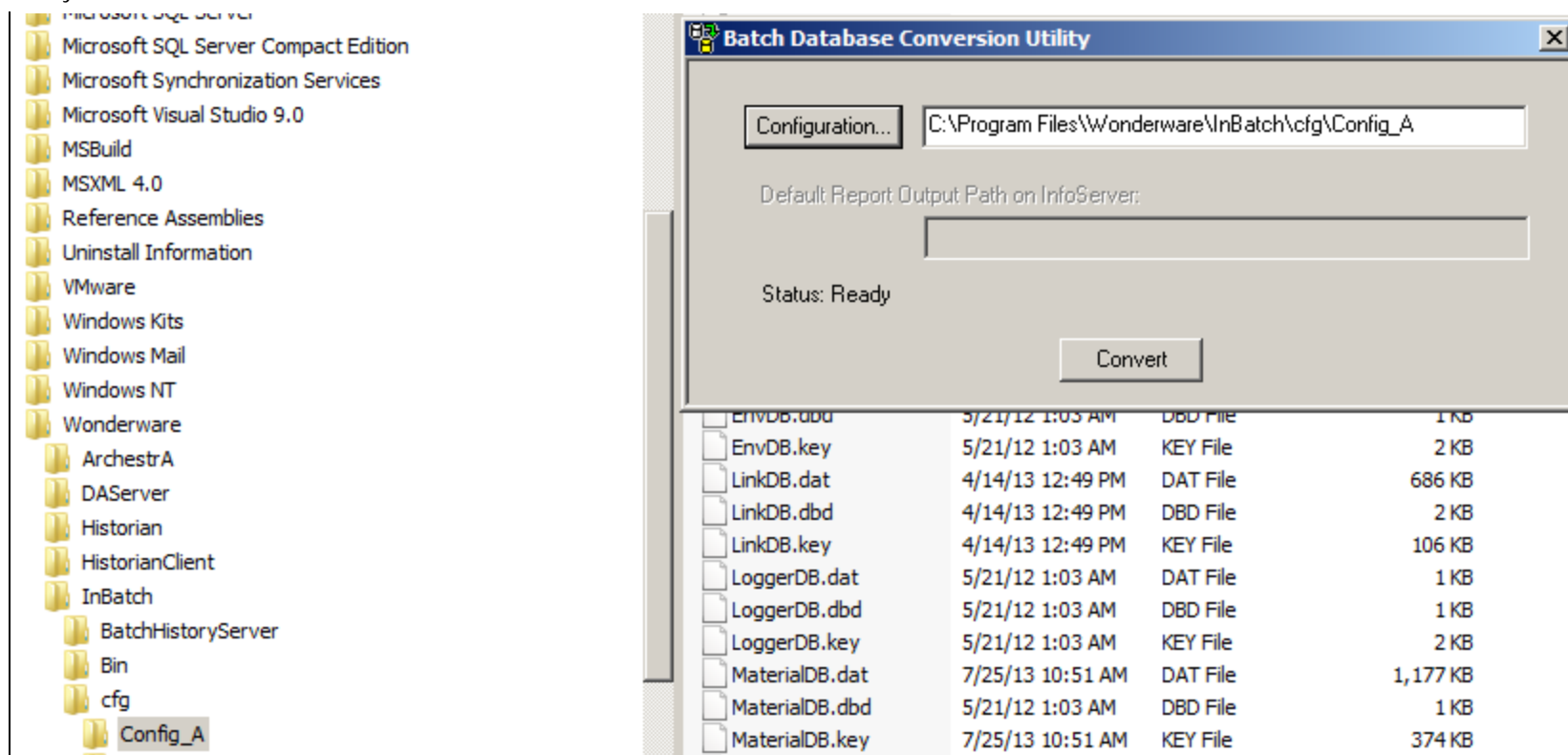


FIGURE 14: BATCH DB CONVERSION UTILITY

8. To check and see that the conversion ran correctly, review the file in config_A called **DBCvt.txt**. This is the conversion record.

Here is the log file content where the conversion was run and was successful:

```

11. 131024.153705 EnvDBCvt100 DBCvt.c 168 ( 0)
12. ***** 8.1-->10.0 CONVERSION INITIATED *****
13. 131024.153705 EnvDBCvt100 DBCvt.c 182 ( 0)
14. 8.1 path: (.\OldDB95\), 10.0 path: (.)
15. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 56 ( 0)
16. ***** Running EnvDBCvt100 *****
17. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 119 ( 0)
18. Entered EnvConvert
19. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 128 ( 0)
20. completed IBMX assignment to Applications struct
21. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 138 ( 0)
22. IBMX already found skipping Migration
23. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 252 ( 0)
24. Entered AddRecipeCompare
25. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 262 ( 0)
26. completed RecipeCompare assignment to Applications struct
27. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 278 ( 0)

```


28. Successfully added RecipeCompare application to Applications table
29. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 210 (0)
30. Entered AddRecipeCompareRuntimeOption
31. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 218 (0)
32. completed RecipeCompare assignment to Runtime struct
33. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 236 (0)
34. Successfully added RecipeCompare application to Runtime table
35. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 281 (0)
36. Successfully migrated
37. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 299 (0)
38. Migrating BatchManager options
39. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 333 (0)
40. Adding Disconnect Clients parameter
41. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 337 (0)
42. Added Disconnect Clients parameter successfully
43. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 333 (0)
44. Adding Batch Stats parameter
45. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 337 (0)
46. Added Batch Stats parameter successfully
47. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 344 (0)
48. Assigning Disconnect Clients parameter to Runtime application
49. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 349 (0)
50. Deleting Batch Stats parameter from Runtime application if found
51. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 471 (0)
52. Remove TIM related Functions
53. 131024.153705 EnvDBCvt100 EnvDBCvt100.c 508 (0)
54. Successfully migrated
55. 131024.153706 EnvDBCvt100 DBCvt.c 238 (0)
56. ***** CONVERSION COMPLETED: ok *****

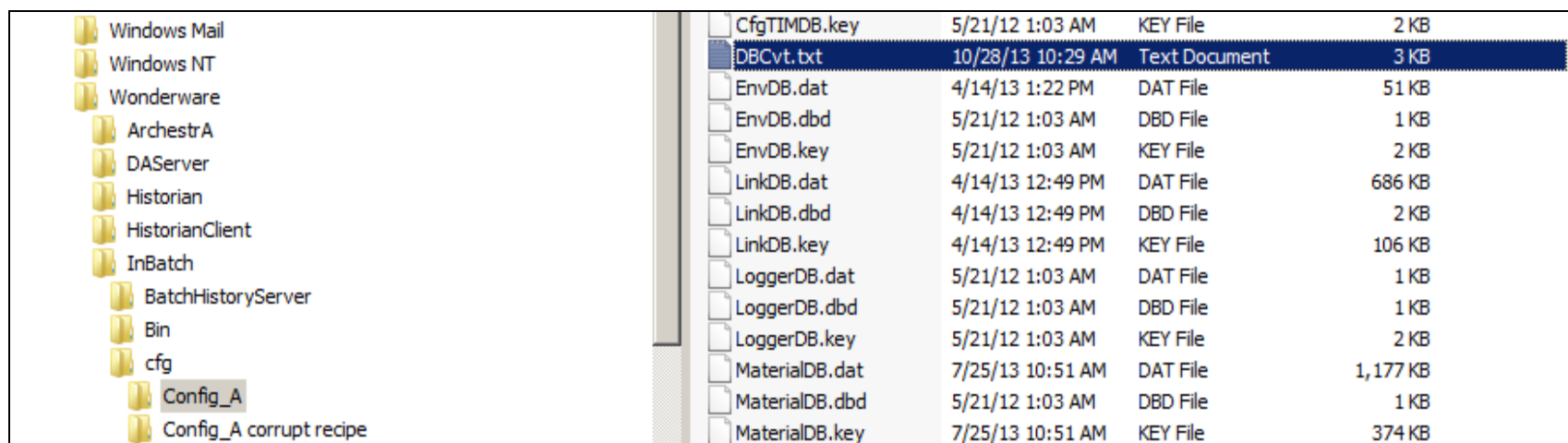


FIGURE 15: DBCVT.TXT CONVERSION LOG FILE

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](#).