Perceptive Content Database Upgrade Package for SQL Server

Reference Guide

Version: Foundation EP2

Written by: Product Knowledge, R&D
Date: March 2021

Hyland™

Copyright

Information in this document is subject to change without notice. The software described in this document is furnished only under a separate license agreement and may be used or copied only according to the terms of such agreement. It is against the law to copy the software except as specifically allowed in the license agreement. This document or accompanying materials contains certain information which is confidential information of Hyland Software, Inc. and its affiliates, and which is subject to the confidentiality provisions agreed to by you.

All data, names, and formats used in this document's examples are fictitious unless noted otherwise. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright law, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Hyland Software, Inc. or one of its affiliates.

Hyland® and Hyland Software®, as well as Hyland product names, are registered and/or unregistered trademarks of Hyland Software, Inc. and its affiliates in the United States and other countries. All other trademarks, service marks, trade names and products of other companies are the property of their respective owners.

© 2020 Hyland Software, Inc. and its affiliates. All rights reserved.

Table of Contents

Copyright **2**
Overview **5**
Benefits **5**
Summary of steps **5**
With uptime 5
Setup steps *5*
Uptime steps *6*
Without uptime 6
Status commands 6
Prerequisites **6**
Create the packages **6**
IN_DB_UTIL package 6
IN_DB_UPGRADE_7500 package 7
Setup steps **7**
IN_UPGRADE_CONTROL 7
Synchronization framework 8
Synchronization steps **8**
Uptime steps **8**
Downtime steps **9**
Pre-downtime steps 9
Downtime steps 9
Post downtime steps 9
Additional synchronization commands **9**
IN_7500_SP_SYNC_RESET 10
IN_7500_SP_SYNC_REMOVE 10
Available parameters **10**
Examples 11
View and update parameter default values **12**
Troubleshooting commands **12**
Clean up after an early exit **13**
Debugging **13**

Overview

The following document guides you through the upgrade process using the new packages, IN_DB_UTIL and IN_DB_UPGRADE_7500.

You can use this package as an alternative to the traditional database incremental scripts.

This collection of procedures and functions (package) is designed to provide a means for you to upgrade your Perceptive Content database schema in a manner that facilitates the execution of the most time consuming schema changes in uptime to reduce the duration of downtime events during database upgrades.

This package supports upgrading the inuser database schema to version 7.5.0.0 (EP2) from 7.4.0.3 or 7.2.3.0 or 7.2.0.0 or 7.1.5.2.

This upgrade package is ideal for customers with very large databases who might benefit from executing the long running operations in uptime, resulting in a much shorter downtime event during the upgrade.

Benefits

- This upgrade package provides for additional flexibility by facilitating an upgrade to 7.5.0.0 from any previous version between 7.1.5.2 – 7.4.0.3.
- The 7.2.2.0 and 7.5.0.0 incremental database scripts include operations that rebuild the IN_DOC table and its indexes and constraints. These operations are time consuming and result in an extended downtime for larger databases. This upgrade package provides a means for accomplishing these lengthy operations during uptime in the days or hours leading up to the downtime and results in a much shorter downtime event.
- The 7.4.0.3 incremental database script includes the rebuild of the indexes on the IN_INSTANCE_PROP table to create them as filtered indexes to improve performance and decrease the size of those indexes. This upgrade package provides a means for accomplishing these index rebuilds in uptime in the days or hours leading up to the downtime event and results in a much shorter downtime event.
- This upgrade package provides you with more control with regard to incrementally completing the uptime steps over short periods of time by providing parameters to limit the duration of uptime steps through the use of the MAX_MINUTES parameter.
- This upgrade package provides you with more control over database upgrade operations by surfacing parameters such as filegroup names, max degree of parallelism (MAXDOP) and fillfactor for both uptime and downtime operations.

Summary of steps

The following section outlines the summary of steps described in this document.

With uptime

Setup steps

Use the following commands to create the IN_UPGRADE_CONTROL table and synchronization framework.

```
EXEC inuser.IN_7500_SP_UPGRADE_SETUP;
```

```
EXEC inuser.IN_7500_SP_SYNC_SETUP;
```

Uptime steps

Use the following commands to execute qualifying uptime schema changes until all steps have completed.

```
EXEC inuser.IN_7500_SP_UPTIME_STEPS;
```

Use the following command to synchronize, as needed, and propragate rows until downtime.

```
EXEC inuser.IN_7500_SP_SYNC_TABLE_IN_DOC;
```

Use the following commands to execute qualifying downtime schema changes.

```
EXEC inuser.IN_7500_SP_DOWNTIME_STEPS;
```

Without uptime

Use the following commands to setup the upgrade and sychronization framework and execute all qualifying uptime and downtime schema changes.

```
EXEC inuser.IN_7500_SP_UPGRADE_SETUP;
EXEC inuser.IN_7500_SP_SYNC_SETUP;
EXEC inuser.IN_7500_SP_DOWNTIME_STEPS;
```

Status commands

Use the following commands to check the synchronization and uptime status.

```
EXEC inuser.IN_7500_SP_SYNC_GET_STATUS;
EXEC inuser.IN_7500_SP_UPTIME_GET_STATUS;
```

Prerequisites

Within SQL Server Management Studio (SSMS) execute the following scripts to create the packages.

Note You must execute these scripts with SQLCMD Mode enabled. To enable SQLCMD mode, from the main **SSMS** toolbar, click **Query** and then select **SQLCMD Mode**.

Create the packages

Within SQL Server Management Studio (SSMS) execute the following scripts to create the packages.

IN_DB_UTIL package

The IN_DB_UTIL package is a collection of integrated procedures and functions that perform common database tasks that were previously part of the database incremental scripts. To create the various database functions and stored procedures used by the upgrade package, complete the following step.

- In SSMS, execute the following script.

```
PerceptiveContentDB_IN_DB_UTIL_SQLServer_Package_v3.0.sql
```

IN_DB_UPGRADE_7500 package

The IN_DB_UPGRADE_7500 package is a collection of integrated procedures and functions that manage the synchronization process and execute database schema changes for both uptime and downtime steps.

Prior to executing this script please review the upgrade and synchronization parameter default values. These values are saved during execution of the IN_7500_SP_UPGRADE_SETUP procedure and are used when populating the IN_UPGRADE_CONTROL table. These default values are used by the various procedures unless otherwise specified during execution.

Please reference the IN_7500_SP_UPGRADE_SETUP usage notes and parameters section for more details.

Note The system does not validate these parameters during package creation but instead validates them later during the execution of each command.

```
-- Upgrade and Synchronization Default Values
:setvar v_filegroup_name 'PRIMARY'
:setvar v_max_minutes    0
:setvar v_max_dop        0
:setvar v_fillfactor     100
:setvar v_debug          'NO'

-- For upgrades starting earlier than 7.5.0.0
:setvar v_sync_batch_size  50000
:setvar v_sync_defer_fk    'NO'
:setvar v_sync_keep_src    'YES'
```

To create the various database functions and stored procedures used to upgrade the INOW schema to version 7.5.0.0 from schema versions 7.1.5.2 through 7.4.0.3, complete the following step.

- In SSMS, execute the following script.

```
PerceptiveContentDB_7500_SQLServer_Package.sql
```

Setup steps

IN_UPGRADE_CONTROL

To create the IN_UPGRADE_CONTROL table, which holds the default parameter values that are used if not otherwise specified when executing the procedures, complete the following step.

- In SSMS or at the command prompt (sqlcmd), execute the following command.

```
EXEC inuser.IN_7500_SP_UPGRADE_SETUP;
```

The following parameters are available to use.

- @FILEGROUP_NAME
- @DEFAULT_MAX_MINUTES
- @DEFAULT_MAX_DOP
- @DEFAULT_FILLFACTOR
- @DEFER_FK_CHECK
- @SYNC_DEFAULT_BATCH_SIZE

- @SYNC_KEEP_SRC_TABLE
- @DEBUG

Synchronization framework

To create the synchronization framework for propagating data between the IN_DOC and NEW_DOC tables, which facilitate the column modifications and data population and index creation of the new version of the IN_DOC table in uptime, complete the following step.

- In SSMS or at the command prompt (sqlcmd), execute the following command.

```
EXEC inuser.IN_7500_SP_SYNC_SETUP;
```

The following parameter is available to use.

- @FILEGROUP_NAME

Synchronization steps

This synchronizes the new version of the IN_DOC table (NEW_DOC) with the existing version (IN_DOC) based on the rows in the staging table (IN_7500_SYNC_STAGE_DOC).

1. In SSMS or at the command prompt (sqlcmd), execute the following command.

```
EXEC inuser.IN_7500_SP_SYNC_TABLE_IN_DOC;
```

The following parameters are available to use.

- @MAX_MINUTES
- @BATCH_SIZE

To view the current status of the synchronization between the IN_DOC and NEW_DOC tables, complete the following step.

2. In SSMS or at the command prompt (sqlcmd), execute the following command.

```
EXEC inuser.IN_7500_SP_SYNC_GET_STATUS;
```

The system displays a status report that includes synchonization metrics and details regarding staged data.

Uptime steps

To execute qualifying uptime schema changes, complete the following step.

Note If you choose not to execute any steps in UPTIME, you can skip this procedure and go directly to the DOWNTIME_STEPS procedure.

- In SSMS or at the command prompt (sqlcmd), execute the following command.

```
EXEC inuser.IN_7500_SP_UPTIME_STEPS;
```

The following parameters are available to use.

- @MAX_MINUTES
- @MAX_DOP
- @FILLFACTOR

You can schedule this procedure or manually execute it as necessary to incrementally execute the uptime steps at any point after upgrade and sychronization setup and prior to the execution of the downtime steps.

To view the current status and progress of the uptime steps execute the following command.

- In SSMS or at the command prompt (sqlcmd), execute the following command.

```
EXEC inuser.IN_7500_SP_UPTIME_GET_STATUS;
```

Downtime steps

Pre-downtime steps

Important Before executing the downtime steps, ensure the following:

- Make sure you have a good full database backup prior to executing the downtime steps.
- Stop all the services for the Perceptive Content Application server.
- Check for any other connections to the database and disconnect them.

Downtime steps

To execute the qualifying downtime schema changes necessary to take the INOW database schema to 7.5.0.0, complete the following steps. This procedure calls the IN_7500_SP_SYNC_TABLE_IN_DOC and IN_7500_SP_UPTIME_STEPS procedures to ensure all prerequisite actions are completed.

- In SSMS or at the command prompt (sqlcmd), execute the following command.

```
EXEC inuser.IN_7500_SP_DOWNTIME_STEPS;
```

The following parameters are available to use.

- @MAX_DOP
- @FILLFACTOR
- @SYNC_KEEP_SRC_TABLE
- @SYNC_DEFER_FK_CHECK

Post downtime steps

Upon successful completion of the DOWNTIME_STEPS, proceed with the following Perceptive Content Server related upgrade steps.

1. Perceptive Content Server related upgrade steps.
2. Full back-up of the database.
3. Update database statistics.
4. Conduct standard user acceptance testing procedures.

Additional synchronization commands

The following procedures can be executed as necessary between IN_7500_SP_SYNC_SETUP and DOWNTIME_STEPS to accomplish certain tasks as needed.

IN_7500_SP_SYNC_RESET

To reset the synchronization framework between the IN_DOC table and the NEW_DOC table, complete the following step.

- In SSMS or at the command prompt (sqlcmd), execute the following command.

```
EXEC inuser.IN_7500_SP_SYNC_RESET;
```

This truncates the NEW_DOC table, drops any indexes on the table that were previously created, resets the IN_7500_SYNC_SEQ_DOC sequence and truncates the IN_7500_SYNC_STAGE_DOC, IN_7500_SYNC_ERROR_DOC, IN_7500_SYNC_STATE_DOC, IN_7500_SYNC_HIST_DOC tables.

IN_7500_SP_SYNC_REMOVE

To remove the synchronization framework between the IN_DOC table and the NEW_DOC table, complete the following step.

- In SSMS or at the command prompt (sqlcmd), execute the following command.

```
EXEC inuser.IN_7500_SP_SYNC_REMOVE;
```

This drops the following database objects which comprise the synchronization framework.

IN_7500_SYNC_TRG_DOC trigger

IN_7500_SYNC_SEQ_DOC sequence

IN_7500_SYNC_STAGE_DOC table

IN_7500_SYNC_ERROR_DOC table

IN_7500_SYNC_STATE_DOC table

IN_7500_SYNC_HIST_DOC table

NEW_DOC table and indexes

Available parameters

When calling the various procedures you can specify one or more of the following parameters based on your preferences. All procedures can be executed without specifying any additional parameters.

Note Not all parameters are available for every procedure.

Paramenter	**Description**
@FILEGROUP_NAME	Specifies the name of the filegroup in which to create the primary objects. The default is either the current filegroup for the object or the default filegroup for the database.
@MAX_MINUTES	Specfies the number of minutes used to define an end time that a synchronization or upgrade task is allowed to start new operations. The default is 0 which is equal to unlimited.
@ MAX_DOP	Specfies the degree of parallelism for the session. Used during initial loading of the destination table and index creation.

	The default is 0.
@ FILLFACTOR	Specfies the percentage to fill data and index pages. Used during the creation of tables and indexes created during uptime and downtime operations. The default is 100.
@DEFER_FK_CHECK	Used to indicate whether the validation of foreign key constraints, to and from the new IN_DOC table, will be deferred and manually executed after the upgrade. The default is NO. The Foreign Key validation is the longest part of the downtime. Deferring that step is useful for facilitating the shortest possible downtime. You can defer the Foreign Key validation by specifying 'YES' ('Y') for the DEFER_FK_CHECK parameter when executing the downtime steps. If you defer the Foreign Key validation then the system displays the validation commands instead of executing them so you can manually execute them after the downtime steps. **WARNING** The Foreign Key check deferral delays the identification of any data consistency issues and could result in time consuming troubleshooting efforts after the upgrade.
@BATCH_SIZE	Used by the IN_7500_SP_SYNC_TABLE_IN_DOC procedure to define the batch size (number of rows) to populate the work queue (cursor) per batch. The default is 50,000 rows.
@KEEP_SRC_TABLE	Specfies whether to keep the original/source table (IN_DOC) upon completion of the downtime steps. The default is YES.
@DEBUG	Specfies whether to display additional logging for debugging purposes. Includes state changes, call stack and other details. The default is NO.

Examples

The following examples show some of the procedures with parameters specified during execution.

Specification of the parameters are not required during execution unless you want to override the default values. If you do not specify a parameter, then the default values are used.

```
EXECUTE inuser.IN_7500_SP_UPGRADE_SETUP
  @FILEGROUP_NAME = 'PRIMARY',
  @DEFAULT_MAX_MINUTES = 60,
  @DEFAULT_MAX_DOP = 4,
  @DEFAULT_FILLFACTOR = 90,
  @DEFER_FK_CHECK = 'NO',
  @SYNC_DEFAULT_BATCH_SIZE = 50000,
  @SYNC_KEEP_SRC_TABLE = 'YES',
  @DEBUG = 'NO';
```

```
EXECUTE inuser.IN_7500_SP_SYNC_SETUP
  @FILEGROUP_NAME = 'PRIMARY';

EXECUTE inuser.IN_7500_SP_SYNC_TABLE_IN_DOC
  @MAX_MINUTES = 10,
  @BATCH_SIZE = 50000;

EXECUTE inuser.IN_7500_SP_UPTIME_STEPS
  @MAX_MINUTES = 1,
  @MAX_DOP = 4,
  @FILLFACTOR=90;

EXECUTE inuser.IN_7500_SP_DOWNTIME_STEPS
  @MAX_DOP = 4,
  @FILLFACTOR = 90,
  @SYNC_KEEP_SRC_TABLE = 'YES',
  @SYNC_DEFER_FK_CHECK = 'NO';
```

View and update parameter default values

You can manually update the IN_UPGRADE_CONTROL table to change the default values. To view the current parameter defaults, complete the following step.

- In SSMS or at the command prompt (sqlcmd), execute the following command.

  ```
  SELECT * FROM inuser.IN_UPGRADE_CONTROL;
  ```

  The following is an example of updating default values.

  Notes

  The values shown below are the only values that should ever be modified in this table. Do not modify any other values.

  The values used in the following example are not the defaults in some cases. See the parameter descriptions above for defaults.

  ```
  UPDATE inuser.IN_UPGRADE_CONTROL SET
  DEFAULT_MAX_MINUTES = 60
  ,DEFAULT_MAX_DOP = 4
  ,DEFAULT_FILLFACTOR = 95
  ,DEFER_FK_CHECK = 'NO'
  ,SYNC_DEFAULT_BATCH_SIZE = 50000
  ,SYNC_KEEP_SRC_TABLE = 'YES'
  ,DEBUG = 'NO';
  ```

Troubleshooting commands

- To view active session details, execute the following command.

  ```
  EXEC inuser.IN_DB_UTIL_SP_SESSIONS_DISPLAY 'ALL';
  ```

- To view call stack details, execute the following command.

  ```
  EXEC inuser.IN_DB_UTIL_SP_CALLSTACK_PRINT 'ALL';
  ```

- To view the contents of the temporary table containing active session details and attributes, execute the following command. This table is queried by the IN_DB_UTIL_SP_SESSIONS_DISPLAY procedure.

```
SELECT * FROM ##IN_DB_UTIL_SESSION_ATTRIBUTES ORDER BY IDENTIFIER;
```

- To view the contents of the temporary table containing call stack depth between stored procedures execute the following command. This table is queried by the IN_DB_UTIL_SP_CALLSTACK_PRINT procedure.

```
SELECT * FROM ##IN_DB_UTIL_CALLSTACK ORDER BY IDENTIFIER, DEPTH;
```

- To view the current state of synchronization, execute the following command. This table is queried by the IN_7500_SP_SYNC_GET_STATUS procedure.

```
SELECT * FROM INUSER.IN_7500_SYNC_STATE_DOC;
```

- To view the history of synchronization activity, execute the following command. This table is queried by the IN_7500_SP_SYNC_GET_STATUS procedure.

```
SELECT * FROM INUSER.IN_7500_SYNC_HIST_DOC ORDER BY SYNC_DATETIME;
```

- To view the history of synchronization errors, execute the following command. This table is queried by the IN_7500_SP_SYNC_GET_STATUS procedure.

```
SELECT * FROM INUSER.IN_7500_SYNC_ERROR_DOC ORDER BY ACTION_DATETIME;
```

Clean up after an early exit

You can use the following procedures to manually clear the call stack and session attribute tables as needed due to an unhandled exception or after manually cancelling an operation before completion.

Important Do not execute any of the following commands if any operations are still in progress as it will clear the call-stack and session attributes which will lead to undesired results and unhandled exceptions.

- Clear session and call stack for all identifiers (SYNC and UPGRADE)

```
EXEC inuser.IN_DB_UTIL_SP_CLEAR_IDENTIFIER 'ALL';
```

- Clear session and call stack for only the SYNC identifier

```
EXEC inuser.IN_DB_UTIL_SP_CLEAR_IDENTIFIER 'SYNC';
```

- Clear session and call stack for only the UPGRADE identifier

```
EXEC inuser.IN_DB_UTIL_SP_CLEAR_IDENTIFIER 'UPGRADE';
```

Debugging

You can use the @DEBUG parameter with most procedures if additional debugging is necessary.

This results in a very verbose execution that contains state change information and call stack details and other background call details to assist with debugging certain issues.

```
@DEBUG = 'YES'
```