Question / Problem:

Is it possible to determine if there are orphan messages stuck in the wsa_messages table?

Answer / Solution:

One of the elements that can contribute to a backlog of data in the TotalAgility_Reporting_Staging, or slower processing of a backlog, is orphan data in the wsa_messages table. In KTA 7.5.0.20 and earlier orphan data may prevent valid data from processing, or prevent it from processing at a normal throughput. To work around this, it may be necessary to tell the reporting service to process a larger batch of messages at a time (larger than the number of orphans). For more details see Reporting processing batch size - WSABatchSize.

Orphan records may still exist in KTA 7.5.0.21+ and KTA 7.6+, but improvements allow continued processing to occur even when orphan data is present. Additionally one cause of orphan records is resolved in 7.5.0.25, 7.6.0.11, 7.7.0.9, 7.8.0.1 (Bug 1496170 - extr_batch_ses_key isn't reported in field_accum_fact table).

As part of troubleshooting a reporting backlog issue, it may be useful to quantify any orphan data present. To do so, stop the reporting service and run the following diagnostic query on the TotalAgility_Reporting_Staging database to show the number of orphan messages, of which there are a few different types.

```sql
WITH MatchedMessages
AS (
    SELECT ISNULL(ba.batch_action_id,be.batch_action_id) AS batch_action_id, ExpectedTotalMsg,
           ISNULL(EditMsgCount,0) AS EditMsgCount, ISNULL(ActionMsgCount,0) AS ActionMsgCount
    FROM (SELECT batch_action_id, COUNT(*) AS ActionMsgCount, SUM(messages_count) AS ExpectedTotalMsg
          FROM wsa_messages
          WHERE function_name='KCBatchAction'
          GROUP BY batch_action_id
        ) AS ba
    FULL OUTER JOIN (SELECT batch_action_id, COUNT(*) AS EditMsgCount
                      FROM wsa_messages
                      WHERE function_name='BatchEdit'
                      GROUP BY batch_action_id
        ) AS be
      ON ba.batch_action_id=be.batch_action_id
    )
```
SELECT MatchStatus, COUNT(*) AS MatchTypeCount, SUM(TotalMessages) AS TotalMessages
FROM (SELECT batch_action_id, EditMsgCount+ActionMsgCount AS TotalMessages,
    CASE
        WHEN ActionMsgCount>1 THEN 'NotExpected-MultipleActionSameId'
        WHEN ActionMsgCount=1 AND (ExpectedTotalMsg IS NULL OR ExpectedTotalMsg <2) THEN
            'NotExpected-ActionNotExpectingEditMessages'
        WHEN ActionMsgCount=0 THEN 'Orphan-ActionMissing'
        WHEN ExpectedTotalMsg> EditMsgCount+ActionMsgCount THEN 'Orphan-
            LessEditMessagesThanExpected'
        WHEN ExpectedTotalMsg= EditMsgCount+ActionMsgCount THEN 'OK-ExpectedMatch'
        ELSE 'Unknown'
    END AS MatchStatus
    FROM MatchedMessages
) AS ms
GROUP BY MatchStatus

Note that it is not feasible to run this query with the reporting service running. When the reporting service is running, the normal processing of the reporting services will result in locking on the wsa_messages table and cause the query to take much longer. Additionally the amount of time the query takes depends on the total number of records in the wsa_messages table.

To use the query in a tenant database of an OPMT system, add either “reportingstaginglive.” or “reportingstagingdev.” (depending on environment) before each occurrence of the table names. The only table name used here is wsa_message.