GENESYS™

User's Guide

Set-based Archiving with MSSQL

4/30/2025

Set-based Archiving with MSSQL

Purpose: This page presents the SQL queries used for set-based maintenance of the UCS database on an MS SQL RDBMS.

Contents

Prerequisites

See Archiving and Pruning the DB for prerequisites. In particular, before using these queries you must first **run archiving from UCS Manager.**

Creating the Database Link

1. Be sure that the DNS name resolves properly to the archive database server. If not, you can add it to the host file; on Windows, for example, this is located at `C:\WINDOWS\system32\drivers\etc\hosts`.

2. To create the DB link execute the following command. Note that the command will return no error, even if a parameter is wrong or the destination host does not resolve correctly.

    ```
    EXEC sp_addlinkedserver @server = N'bsgenucsdbarch', @srvproduct=N'SQL Server'
    ```

    > ### Important
    >
    > The queries presented here describe the minimum needed to create the database link between the main and archive UCS databases. Depending on the configuration of your database, you may need to pass more parameters, such as `usernames`, `password`, `schemas`, `tablespaces`, and so on. Consult your RDBMS documentation for guidance.

3. To test if creation was successful, execute the following command:

    ```
    select count(*) from bsgenucsdbarch.UCSARCH.dbo.interaction;
    ```

    In this example,

    -   `bsgenucsdbarch` is the destination host.
    -   `UCSARCH` is the database.
    -   `dbo` is the schema.

    Edit these names to match your configuration. Do the same in the queries provided in Moving the Data to the Archive Database below.

4. To drop the link, execute the following command:

    ```
    EXEC sp_dropserver 'bsgenucsdbarch', null;
    ```

    The link can be kept permanently and will not affect UCS operations. But when the link is no longer used, you may wish to drop it for security concerns.

Moving the Data to the Archive Database

1. Use the following commands:

```
insert into bsgenucsdb.UCSArch.dbo.interaction select * from interaction_arch;
 insert into bsgenucsdb.UCSArch.dbo.emailin select * from emailin_arch;
 insert into bsgenucsdb.UCSArch.dbo.emailout select * from emailout_arch;
 insert into bsgenucsdb.UCSArch.dbo.phonecall select * from phonecall_arch;
 insert into bsgenucsdb.UCSArch.dbo.callback select * from callback_arch;
 insert into bsgenucsdb.UCSArch.dbo.chat select * from chat_arch;
 insert into bsgenucsdb.UCSArch.dbo.ixncontent select * from ixncontent_arch;
 insert into bsgenucsdb.UCSArch.dbo.ixnContentSentReceived select * from
 ixnContentSentReceived_arch;
 insert into bsgenucsdb.UCSArch.dbo.document select * from document_arch;
 insert into bsgenucsdb.UCSArch.dbo.cobrowseurl select * from cobrowseurl_arch;
 insert into bsgenucsdb.UCSArch.dbo.attachment select * from attachment_arch;
```

2. If the move is successful, the temporary tables can be dropped:

```
drop table interaction_arch;
 drop table emailin_arch;
 drop table emailout_arch;
 drop table phonecall_arch;
 drop table callback_arch;
 drop table cobrowseurl_arch;
 drop table chat_arch;
 drop table attachment_arch;
 drop table ixncontent_arch;
 drop table ixnContentSentReceived_arch;
 drop table document_arch;
```

End

Archiving is now complete. Return to Archiving and Pruning the DB for descriptions of limitations and failure recovery methods.