Best Practice: Copying a Live SYSPRO Company

SYSPRO 8

Published: May 2022

CONTENTS

Best Practice: Copying a Live SYSPRO Company

Best Practice: Copying a Live SYSPRO Company

The process of copying of an existing live company in SYSPRO is a high-risk activity, as you can accidentally change or transact against your original live company if you get the process wrong. However, as the need for this does arise from time to time, this article provides you with important information related to the process of copying a live company to a test company within your current ***SYSPRO 8*** environment.

Document Objectives and Audience

This document is targeted at personnel and partners working in a ***SYSPRO 8*** support environment. The aim is to provide the SYSPRO community with guidelines on how to handle the copying of an existing SYSPRO company in ***SYSPRO 8***, considering related information that may be relevant and helping to reduce the risk.

Possible Scenarios

Some scenarios that might require the copying of an existing live company in ***SYSPRO 8*** include, but are not limited to:

- Using the copied company for training purposes
- Creating a TEST version of your live company to view possible ramifications of a proposed change
- Creating an archive or repository of the live company as of the date it was copied

Disclaimer

This article provides guidelines and good practices when copying a SYSPRO company. However, your processes, hardware, software, networking, integration, configuration, and other related issues may require additional processes or management over-and-above the items mentioned here.

Additional considerations:

- The content relates to the copying of a company in ***SYSPRO 8*** only and doesn't cover earlier SYSPRO versions.
- This topic *doesn't* cover how to copy or move an *entire environment* (i.e. system and company databases).
- The steps, and SQL script, within this topic only cover the copying of a self-contained SYSPRO company database.

> ⓘ A self-contained SYSPRO company is one that doesn't use another company(s) database for it's Inventory and General Ledger.

Process: Copying a Live Company

As mentioned already, the process to copy a live SYSPRO company carries with it a level of risk. Therefore, the following process covers the steps required for the successful copy of an existing company in ***SYSPRO 8***:

1. Live Company Database Back up

Its important that you backup your company, system-wide and any other relevant databases prior to making any changes or following the guidelines in this article. Therfore, use your normal full backup processes and verify that the backup is successful, as some of these processes will be performed against your live SQL Server Instance using a SQL user with high authority, and any mistakes could have serious consequences.

Use the following steps to backup the live company database, which will be restored later:

1. Open **Microsoft SQL Server Management Studio**, logging in with user credentials that permit database backup and restore functionality.
2. Navigate to and select the applicable company database of the live company that you want to copy.
3. Right-click the database and select the **Back Up** option from the **Tasks** menu.
4. Use the **Add** function to indicate the back up destination.
5. Select **OK** to perform the back up.

A system message is displayed once the back up has completed successfully.

2. New Company Database Restore

After the database has been backed up, restore this as the new company database:

1. Within **Microsoft SQL Server Management Studio**, navigate to and select the **Databases** group folder.
2. Right-click the folder and select the **Restore Database** option.
3. Proceed as follows within the **General** tab of the **Restore Database** screen:
   a. Select the **Device** radio button to indicate this as the **Source**.
   b. At the **Source** > **Database** field, search for and select the backup file that you created previously (e.g. `Syspro8CompanyEdu1.bak`).

      The **Destination** section is then automatically populated based on your selection.
   c. Change the **Database** name in the **Destination** section to that of the new company database that you want to create (e.g. `Syspro8CompanyTest`).
4. Select **OK** to perform the database restore.

A system message is displayed once the restore has completed and the new database then appears within the list of databases.

3. New Company Database Configuration

The newly created database requires some additional configuration before it is ready for use:

Logical File Name

Proceed as follows to change the **Logical file name** for this database:

1. Within **Microsoft SQL Server Management Studio**, navigate to and select the newly created database (e.g. `Syspro8CompanyTest`).
2. Right-click the database name and select **Properties** from the displayed menu.
3. From the **Database Properties** screen, select the **Files** tab.

   At this point, the **Logical Names** will still match the names of the database before it was restored.
4. Change the **Logical Names** in the grid to match the new database names.
5. Select **OK** to apply your changes.

SQL Script: Company ID Updates

This script takes into account all columns within the database that are companies and performs the updates accordingly as follows:

- All company IDs are updated in the company database to the new company ID.
- If applicable (i.e. the company you are copying shares its inventory or general ledger data with another company) the shared inventory and combined GL company IDs in the company control record are updated from the old company ID to the new one.
- If applicable, the records in the GL tables are changed to have the new company ID in place of the old one.

This information relies on the Data Dictionary and custom form definitions which may have Company columns defined as part of the key.

Script: Company ID Updates

```
/*
This script is used to change all company ID's in a company Database.
It is most commonly used when creating a test company from your live company

Use extreme caution here. Ensure that, when making changes, you select the newly
created test company and not the live company.

${systemdb}  - replace this with the name of your system database
${companydb} - replace this with the name of the new company db

*/


SET NOCOUNT ON
DECLARE @SystemDatabase varchar(20) = '${systemdb}'   -- This is your system database
DECLARE @NewDatabase    varchar(20) = '${companydb}'  -- This is the NEW database
DECLARE @OldCompany     char(4) = 'OLD'               -- This is the original company ID in upper case
DECLARE @NewCompany     char(4) = 'NEW'               -- This is the new company ID
DECLARE @SysproOnly     char(1) = 'N'                 -- Set to 'Y' to only update standard SYSPRO tables
DECLARE @SQL            varchar(max)                  -- SQL statements to be executed

-----------------------------
-- NO CHANGES FROM THIS POINT
-----------------------------
-- Ensure company ID's are upper case
SET @OldCompany = UPPER(@OldCompany)
SET @NewCompany = UPPER(@NewCompany)
SET @SysproOnly = UPPER(@SysproOnly)

-- Create a table with the entries that will require checking and updating
IF OBJECT_ID (N'tempdb..#ColumnsToUpdate', N'U') IS NOT NULL
BEGIN
    DROP TABLE #ColumnsToUpdate
END

CREATE TABLE #ColumnsToUpdate
(
TableName varchar(128) COLLATE DATABASE_DEFAULT,
ColumnName varchar(128) COLLATE DATABASE_DEFAULT
)

-- Build up the script to populate the above table.
-- This is done in this fashion so that you only have to define the databases as variables
SET @SQL =
    ';WITH CTE_Inputs (TableName, ColumnName) AS ('

-- If including tables not in the DD or custom form definitions then look for company columns...
IF (SELECT @SysproOnly) = 'N'
BEGIN
SET @SQL = @SQL +
      'SELECT
```

```
        TABLE_NAME  COLLATE DATABASE_DEFAULT,
        COLUMN_NAME COLLATE DATABASE_DEFAULT
      FROM [' + @NewDatabase + '].INFORMATION_SCHEMA.COLUMNS
      WHERE COLUMN_NAME = ''Company'' AND DATA_TYPE LIKE ''%char'' AND CHARACTER_MAXIMUM_LENGTH = 4
      UNION '
END

SET @SQL =  @SQL +
     'SELECT
         a.TableName COLLATE DATABASE_DEFAULT,
         a.ColumnName COLLATE DATABASE_DEFAULT
      FROM [' + @SystemDatabase + '].DdsColumns a
    INNER JOIN [' + @SystemDatabase + '].DdsTables b ON (a.TableName = b.TableName)
    WHERE a.ColumnDataType = ''Company'' AND b.Company IN ('' '',''G'',''I'') AND b.DataId != ''_VIEW''
    UNION
    SELECT
       a.TableName COLLATE DATABASE_DEFAULT,
       b.ColumnName COLLATE DATABASE_DEFAULT
    FROM [' + @SystemDatabase + '].DefnCustomForm a
    INNER JOIN [' + @SystemDatabase + '].DdsColumns b ON ((SUBSTRING(a.TableName,1,LEN(a.TableName) - 1)) = b.TableName)
    INNER JOIN [' + @SystemDatabase + '].DdsTables c ON (b.TableName = c.TableName)
    WHERE a.TableName != '''' AND b.KeyFlag = ''K'' AND b.ColumnDataType = ''Company'' AND c.Company IN (''
'',''G'',''I'')
    AND EXISTS(SELECT * FROM [' + @NewDatabase + '].INFORMATION_SCHEMA.TABLES
               WHERE TABLE_NAME = a.TableName COLLATE DATABASE_DEFAULT)
    )
    INSERT INTO #ColumnsToUpdate (TableName, ColumnName)
    SELECT TableName, ColumnName FROM CTE_Inputs'


BEGIN TRY
    PRINT (@SQL)
    EXEC (@SQL)

    DECLARE @TableName   varchar(128)
    DECLARE @ColumnName  varchar(128)
    DECLARE Update_Cursor CURSOR FOR
        SELECT TableName, ColumnName FROM #ColumnsToUpdate
    OPEN Update_Cursor
    FETCH NEXT FROM Update_Cursor INTO @TableName, @ColumnName
    WHILE @@FETCH_STATUS = 0
    BEGIN
        BEGIN TRANSACTION
        SET @SQL = 'UPDATE [' + @NewDatabase + '].[' + @TableName + '] ' +
                   'SET [' + @ColumnName + '] = ''' + @NewCompany + ''' WHERE [' + @ColumnName + '] =
''' + @OldCompany + ''''
        PRINT @SQL
        EXEC (@SQL)
        COMMIT TRANSACTION
        FETCH NEXT FROM Update_Cursor INTO @TableName, @ColumnName
    END
    CLOSE Update_Cursor
    DEALLOCATE Update_Cursor

    -- Update the AdmCompanyCtl blob
    BEGIN TRANSACTION
    SET @SQL = 'UPDATE [' + @NewDatabase + '].[AdmCompanyControl] ' +
               'SET DataString = ' +
               ' SUBSTRING(DataString,1,120) + ''' +
               @NewCompany +
               ''' + SUBSTRING(DataString,125,28) + ''' +
               @NewCompany +
               ''' + SUBSTRING(DataString,157,820) + ''' +
               CAST(@NewDatabase as char(20)) +
               ''' + SUBSTRING(DataString,997,2) ' +
               'WHERE RecordType = ''00'''
    PRINT (@SQL)
    EXEC (@SQL)
    COMMIT TRANSACTION
    SELECT [UpdateStatus] = 'Completed', [Notes] = 'Company ID changed from ''' + @OldCompany + ''' to
''' + @NewCompany + ''''
END TRY

BEGIN CATCH
    -- Rollback update being performed
    IF @@TRANCOUNT > 0
    ROLLBACK TRANSACTION;

    SELECT [UpdateStatus] = 'Update failed. Please note the below error message when giving feedback to support.'
    -- Error retrieval routine
    SELECT      [ErrorNumber] = ERROR_NUMBER()
            , [ErrorSeverrity] = ERROR_SEVERITY()
            , [ErrorState] = ERROR_STATE()
            , [ErrorProcedure] = ERROR_PROCEDURE()
            , [ErrorLine] = ERROR_LINE()
            , [ErrorMessage] = ERROR_MESSAGE();
END CATCH

DROP TABLE #ColumnsToUpdate
```

Certain entries within the script need to be changed before it can be executed. Therefore, the following details these required updates:

1. Open the Script.

   You can copy the script and paste it directly within a new query in **Microsoft SQL Server Management Studio**.

   Alternatively, you can copy it into a text editor (e.g. **Notepad++**) and save the file with a `.sql` suffix. This enables you to automatically load the SQL script in **Microsoft SQL Server Management Studio** by double-clicking the file.

2. Update the following entries at the top of the script:

   - **@SystemDatabase**

     Change the ${systemdb} to be the SYSPRO system database (e.g. SysproDb).

   - **@NewDatabase**

     Change the ${companydb} to be that of the company database you have just restored (e.g. Syspro8CompanyTest)

   - **@OldCompany**

     Change the 'OLD' entry to the original company ID from which you copied the database (e.g. EDU1)

   - **@NewCompany**

     Change the 'NEW' entry to the company ID of the new database (e.g. EDU2)

   - **@SysproOnly**

     - Set this entry to 'Y' if you only want the update performed against the standard SYSPRO tables.

     - Set this entry to 'N' if you want the update performed against both standard SYSPRO tables and any other tables in the database that contain columns named `Company` which are 4 characters long.

FOR EXAMPLE:

```
SET NOCOUNT ON
DECLARE @SystemDatabase varchar(20) = 'SysproDb'   -- This is your system database
DECLARE @NewDatabase    varchar(20) = 'Syspro8CompnayTest'  -- This is the NEW database
DECLARE @OldCompany     char(4) = 'EDU1'               -- This is the original company ID in upper case
DECLARE @NewCompany     char(4) = 'EDU2'               -- This is the new company ID
DECLARE @SysproOnly     char(1) = 'Y'                 -- Set to 'N' to change non-SYSPRO tables Company
columns
DECLARE @SQL            varchar(max)                  -- SQL statements to be executed
```

3. Execute the script.

   The **Results** tab displays a message indicating that this has completed and confirms the company ID that was changed.

   The **Messages** tab contains the number of records that were changed in each table.

⚠ This script doesn't change any company IDs used within the GL report writer report definitions, nor does it check or change any user defined objects (e.g. stored procedures) which may reference company IDs.

4. Update System Company Database with New Company

The next step is to add the new company to the SysproAdmin table:

1. Open **Microsoft SQL Server Management Studio**.
2. Within the **Object Explorer**, locate the SYSPRO 8 system-wide database (e.g. **Syspro8db**).

> View the `SQLDBN=` entry in your `IMPACT.INI` file to confirm the correct database name.

3. Expand this database, followed by it's **Tables** folder.
4. Right-click the SysproAdmin entry (i.e. the last table in the list) and select the **Edit Top 200 Rows** function from the displayed menu.

> This should already contain an entry for the live company that you are copying from, as well as any other companies configured on your system.

5. Add a new entry row for the new test company and enter the applicable names within the **Company** and **DatabaseName** columns.

FOR EXAMPLE:

Company	DatabaseName	CollationName	TimeStamp
EDU1	Syspro8CompanyEdu1	Latin1_General_BIN	<Binary data>
EDU2	Syspro8CompanyTest	Latin1_General_BIN	<Binary data>
NULL	*NULL*	*NULL*	*NULL*

6. Move off this entry so that the system records it.
7. Exit **Microsoft SQL Server Management Studio**.

Your new test company is now included within the system company database.

5. New Company License Verification

1. Open ***SYSPRO 8*** and enter the relevant operator code in the **User name** field.
2. Browse on the **Company** field.

   The company that you are copying from, and the company that you are copying to, should both be present. The **Reference** for the new company is brought through from the original company.
3. Select the new company, followed by the **Change** function.
4. Update your license for the new company:

   This step is dependent on which Release of ***SYSPRO 8*** you are using:

   SYSPRO 8 2019 R2 and Prior:

   a. From the **Registration** tab, select the **Import License** button to re-import the license using the wizard.

   b. Once the license has been imported, select the **Save** function.

   SYSPRO 8 2020 R1 onwards:

   a. On the **Registration tab**, place the cursor on the **Registration number** and tab off the field to validate your license number.

   b. Select the **Save** function.
5. Back on the **Companies** screen, highlight and select the new company.
6. Select the **Login** button to proceed.

6. New Company's SYSPRO Configuration

1. Once logged into SYSPRO, open the **Setup Options** program (*SYSPRO Ribbon bar > Setup*).
2. Navigate to the **Company Address** section (*Setup Options > Company > Address*):

   a. Update the **REFERENCE** field to an entry that indicates this is your test company (e.g. `This is my test company`).
3. Navigate to the **Company Options** section (*Setup Options > Company > Options*):

   a. Within the **SHARED COMPANIES** section, ensure that the **LEDGER FILE COMPANY** and **INVENTORY FILE COMPANY** fields reflect the company ID of the new test company (as these would have been updated by the script run previously).
4. Save your changes.
5. Exit SYSPRO.

Your test company is now ready for use.

If you browse on your companies when logging into SYSPRO, you can now easily identify which company is which.

Considerations

Please be aware of the following when copying a live SYSPRO company:

- Certain functionality (such as Electronic Signatures and Search Configurations) can be set up at different levels in SYSPRO (i.e. System, Company, Role, Operator).

  If they are configured at **Company level** then these are *NOT* copied.

  However, if these are defined at **System level**, then they will be applied in the new test company.

  > It is important to be aware of this as you may not want some of this functionality in your test environment.
  >
  > For example, Electronic Signatures that are triggered during transaction processing could send an email inappropriately.

FAQs

How is Shared Inventory and Shared General Ledger handled within the SQL databases?

This section is purely informational as the company copy process detailed above only caters for copying a self-contained company database whose Inventory and General Ledger data is contained within its own databases.

Shared Inventory

Its possible with SYSPRO that the inventory from one company can be shared with other companies.

To enable this to happen, there is a shared inventory company ID in the company control record which defaults to the current company when adding a new company.

If you don't change the Inventory company ID during the company copy process, then transactions will be processed against the original company if you post any inventory transactions in the copied company.

Shared General Ledger

Its also possible to consolidate multiple company’s General Ledgers into one database.

The company control record contains the company ID of the company that must contain this company’s GL. When adding a new company, this defaults to the current company (in the same way as the shared inventory).

In addition, all the records of certain GL tables (e.g. GenMaster) are prefixed with the company ID. This is to ensure (if it is shared) that the posting processes uses the correct company ID to retrieve and apply the data that is relevant to that company.

SYSPRO™

www.syspro.com