

SYSPRO 8 File System Reduction

SYSPRO 8

Published: March 2023



CONTENTS

Reduced file system usage

Overview	1
Data Categories Uploaded to SQL Server	2
SYSPRO 8 2022	2
SYSPRO 8 2021 R2	3
SYSPRO 8 2021 R1	6
SYSPRO 8 2020 R2	11
SYSPRO 8 2020 R1	21
Related Information	22

Reduced file system usage

This article explains how **SYSPRO 8** has reduced the files being persisted to the file system (by migrating them to **Microsoft SQL Server**) since its first release.

It also provides detailed information on each data category uploaded to SQL, in which release the change took place and whether the change was optional or mandatory.

Overview

In **SYSPRO 8**, most data and configuration information is stored within **Microsoft SQL Server**.

We are on a journey to reduce file system persistence to a minimum, using SQL Server to store all data and configuration information. This offers many benefits, including:

- Simplifying your backup strategy
- Improving performance
- Protecting your data
- Providing controlled access to third parties (where relevant)

This means that customers migrating from earlier versions of SYSPRO will have all data stored in their `\Data` folder and most from their `\Work` folder uploaded to **Microsoft SQL Server** during the upgrade to **SYSPRO 8**.

Because of this, you will notice an increase in the size of your database, with the increase being proportional to the size of the data being migrated from the file system.



This document relates to **SYSPRO 8**. However, in **SYSPRO Cloud ERP** environments there are no files persisted to the file system – all items can be considered Mandatory and there is nothing for an end-user to do.

Data Categories Uploaded to SQL Server

SYSPRO 8 2022

Customized VBScripts

As part of the **SYSPRO 8** journey to reduce files being persisted to the file system, you can now store your customized VBScripts (usually stored in the `\Work\VBScripts` folder) within the `SysSettings` table of the system-wide database.

You'll need to enable the **PERSIST OPERATOR UI SETTINGS TO SQL** setup option (*Setup Options > System Setup > System-Wide Personalization*) to cater for this relocation.

This ensures that customization can be retrieved seamlessly in the event of a system crash, or loss of data on the file system, or a pod loss in a **SYSPRO Cloud ERP** environment.

You may notice an increase in the size of your database once you update to **SYSPRO 8 2022**; the increase in size being proportional to the size of the data being migrated from the file system.

Advantages of moving this data from the file system to the database includes:

- Centralized access to all related data
- Reduction of file system usage and maintenance
- Simplified portability of data and backup strategy
- Improved performance

Migration and Storage Process

When an operator logs out of SYSPRO and the **PERSIST OPERATOR UI SETTINGS TO SQL** option is enabled, the system checks if the VBScripts and modules are saved to the database by confirming if the `version.txt` file exists in the `\Work\VBScripts` folder.

If this file doesn't exist, then the system assumes that the scripts are not stored in the database and creates this `version.txt` file with a value. The version number is then used to save the scripts and modules into the database.

Upon further logins to SYSPRO, the VBScripts are only retrieved from **Microsoft SQL Server** if the version in the database is higher than the version on the file system.

When an operator performs changes to any VBScripts within SYSPRO using the **VBScript Editor**, the scripts are then immediately saved to the database, ensuring that the latest scripts are in stored in **Microsoft SQL Server**.

SYSPRO 8 2021 R2

Role customization across any machine

To provide operators within defined roles with the same customization experience across any machine (and in the event of a pod loss in a **SYSPRO Cloud ERP** environment) the following role customizations are now retained in SQL within the system-wide database:

- Role customization files (e.g. `\Settings\Role_xxx`, stored in the [SysRoles](#) table)
- Role menu files (stored in the [SysMenu](#) table)

When operators within a defined role log into SYSPRO, the system checks the [SysRoles](#) and [SysMenu](#) tables for any customized role settings or menus defined and returns these if found. If these settings are not found during the start-up process, the system checks the file system (`\Base\Settings`) and, if found, copies these to the respective SQL database tables and returns the saved settings to the SYSPRO instance.

Advantages of having this data available from the database include:

- Centralized access to all related data
- Simplified portability of data and backup strategy
- Improved performance
- Provision for enhanced scalability in future releases

Considerations:

- Storing these settings in SQL is *not* dependent on the **PERSIST OPERATOR UI SETTINGS TO SQL** system-wide setup option being enabled (*Setup Options > System Setup > System-Wide Personalization*).
- Although role customization settings are retained within the `\Base\Settings` folder of the file system, these files are updated from the SQL database whenever changes are made to the role customization settings in SYSPRO, followed by a graceful exit of SYSPRO.

Operator customization across any machine

To provide operators with the same customization experience across any machine (and in the event of a pod loss in a **SYSPRO Cloud ERP** environment) the following customization settings per operator are now retained in SQL within the **SysSettings** table of the system-wide database:

- System-wide personalization settings
- Recent Programs
- Favorites
- Other SYSPRO menu personalization

When an operator logs into SYSPRO and the **PERSIST OPERATOR UI SETTINGS TO SQL** option is enabled, the system checks the SQL system database for any customized operator settings defined and returns these if found. If these settings are not found during the start-up process, the system checks the file system (`\Base\Settings`) and, if found, copies these to the **SysSettings** table and returns the settings to the SYSPRO instance.

Advantages of storing this data in the database include:

- Centralized access to all related data
- Simplified portability of data and backup strategy
- Improved performance
- Provision for enhanced scalability in future releases

Considerations:

- Storing these settings in SQL is dependent on the **PERSIST OPERATOR UI SETTINGS TO SQL** system-wide setup option being enabled (*Setup Options > System Setup > System-Wide Personalization*).
- Although the operator's personalized settings are still retained within the `\Base\Settings` folder of the file system, this file is updated from the SQL database whenever an operator makes changes to their operator settings in the **Setup Options** program and exits SYSPRO gracefully.

Operator settings now retained in SQL:

- Recent Programs
- Favorites
- SYSPRO menu personalization
- System-wide personalization settings (*Setup Options > System Setup > System-Wide Personalization*):

Forms:

- - Address formatting
 - Override font
 - Font
 - Show captions in bold
 - Show icons for custom fields
 - Default numeric alignment
 - Always apply numeric alignment
 - Show description of keys in caption
 - Highlight editable form fields
 - Highlight color
 - Map URL to use for directions
 - Map URL
 - Google API key
 - Use short names when resolving addresses

Workspace:

- Use Predictive Search
- Show toolbar navigation buttons
- Enable Task Dialog customization
- Enable Toast Notifications
- Toast Notifications delay
- Close Toast Notification when typing
- Web Browser emulates highest IE version
- Save primary window settings upon exit
- Validate G/L sections as you type
- Display G/L sections as drop-down list
- Launch help from status bar
- Show system-wide view when logging in
- Show role view when logging in
- Enable roaming users
- Enable advanced browse

SYSPRO 8 2021 R1

SRS Document Printing

During the **SYSPRO 8 2020 R2** release, the *custom* SRS Document Print XML control files were migrated to **Microsoft SQL Server** as part of the initiative to reduce file system usage.

Continuing on this journey, the setup and printing processes for SRS Document Printing is now optimized to reduce file persistence and address the overhead of loading the control information into memory when using SRS Document Printing.

The *standard* SRS Document Print XML control files and templates are now loaded into a set of three global SQL temporary tables for each instance of SYSPRO as they are required:

- Standard document types
- Standard document samples
- Available archive fields

In addition, the SRS Document Print programs that read the control files are now SQL-optimized for faster retrieval and updating of data. This improves performance, eliminates the use of in-memory collections and reduces the scope of resource locks.



The scope of the resource lock is now reduced to only lock the document type that is being maintained and the document that is being maintained or designed (i.e. multiple users can now maintain different document types and different documents at the same time).

RTF and CMS

As part of the journey to reduce file system persistence, when running **SYSPRO 8 on-premise** you can opt to retain the following RTF and CMS data in the SQL database (instead of the file system):

- RTF notepad documents attached to entities
- CMS Activity body RTF documents
- CMS Activity attachments
- CMS Organization lists



This functionality is available from the new **Migrate Files To SQL** program (*Program List > Administration > General Setup*) that lets you migrate these files to your SQL database (and vice versa). You can also run the program from the **Migrate Files to SQL** option from the **Functions** menu of the **Setup Options** program.



You can migrate these files to your SQL database and vice versa using the **Migrate Files To SQL** program (*Program List > Administration > General Setup*).

Alternatively, you can access this program via the **MIGRATE RTF FILES TO SQL FROM FILE SYSTEM** or **MIGRATE RTF FILES FROM SQL TO FILE SYSTEM** hyperlink against the **RTF AND CMS FILES IN SQL** option in the **Setup Options** program (*Setup Options > System Setup > System-Wide Personalization*). The name of this hyperlink changes depending on the current state of your file storage.

Advantages of moving this data from the file system to the database includes:

- Centralized access to all related data
- Reduction of file system usage
- Simplified portability of data and backup strategy
- Improved performance
- Provision for enhanced scalability in future releases

File locations versus SQL tables:

- **RTF notepad documents attached to entities**

File system location: `\Work\Notes`

SQL Database: `{systemdb}` - SQL table: `AdmNotepadData`

- **CMS Activity attachments**

File system location: `\Work\crm_{compid}\activity\attach`

SQL Database: `{companydb}` - SQL table: `CrmActAttachData`

- **CMS Activity body RTF documents**

- File system location: `\Work\crm_{compid}\activity\body`
SQL Database: `{companydb}` - SQL table: `CrmActBodyData`
- **CMS Organization lists**
File system location: `\Work\crm_{compid}\list`
SQL Database: `{companydb}` - SQL table: `CrmOrgListDefn`

Migrating data from the file system to the database



To ensure that the process completes correctly, the **Migrate Files To SQL** program must be run without any users being connected or processing related documents.

When you run the **Migrate Files To SQL** program to migrate the data from your database to **Microsoft SQL Server**, you are prompted to confirm that you want to start the conversion and, once confirmed, the process proceeds as follows:

1. RTF Notes are migrated into the database:
 - a. A backup folder named `Backup_ccyy_mm_dd` is created within the `\Work\Notes` folder on the file system.
 - b. The system retrieves each entry of the `AdmNotepad` table that's flagged as not residing in SQL, records the file name defined and inserts the contents of this file into the `AdmNotepadData` table of the system-wide database.

2. CMS files are migrated into the database:



This process is repeated for each company with the **Contact Management System** module installed.

- a. The system verifies the following:
 - i. The **Contact Management System** module is installed for the company.
 - ii. The data files have not yet been migrated to **Microsoft SQL Server**.



This check allows for a restart if the process previously failed.

- b. Backup folders named `Backup_ccyy_mm_dd` are created within the `\Work\crm_{compid}\activity` and `\Work\crm_{compid}\list` folders on the file system.
- c. The migration of the CMS activity bodies and attachments proceeds as follows:
 - i. The system retrieves each entry of the `CrmActivity` table that's flagged as not yet residing in SQL and writes information regarding the body and any attachments linked to the activity to a temporary file.

- ii. The contents of each file are uploaded to the [CrmActBodyData](#) and [CrmActAttachData](#) tables once all entries have been written to the temporary file.



If an activity has multiple attendees, then a copy of the body and attachment rows is written for each attendee. Therefore, the files are only moved to the backup folders once everything has been uploaded.

- d. The migration of the CMS organization lists proceeds as follows:
 - i. The system opens each of the organization lists found in the `\Work\crm_{compid}\List` folder and extracts the information (including the operator, description, etc.).
 - ii. The files are uploaded, and the extracted information is then written to the relevant row in **Microsoft SQL Server**.
3. The relevant rows in the [AdmNotepad](#) and [CrmActivity](#) tables are updated to indicate that the data now resides in **Microsoft SQL Server**.
4. The files are moved to the respective backup folders once they have successfully uploaded.
5. Upon exiting the program, an entry is written to the [AdmSystemAuditLog](#) table indicating if the migration was successful or not.

Migrating data from the database back to the file system

If you later decide that holding the RTF and CMS data in **Microsoft SQL Server** is not suitable for your environment (e.g. third party applications require access to the files) then you can migrate this data back to the file system by running the **Migrate Files To SQL** program again.

New environments that require RTF and CMS data to be stored in SQL Server

New on-premise environments will store the RTF and CMS data on the file system by default.

Therefore, if you configure a new SYSPRO environment that requires the data to reside in **Microsoft SQL Server**, you must run the **Migrate Files To SQL** program to perform this migration.

Diagnostic Data

To assist service personnel in providing speedy resolutions, the diagnostics capability of **SYSPRO 8** has been extended to record a log of identified problems in SQL Server for current and future analysis. A *Black-box* capability has also been introduced to record what was happening on a system prior to failure. Recording the information in SQL is vital for the **SYSPRO Cloud ERP** offering where the file system cannot be used to persist information.

SYSPRO Run Time Errors

When the SYSPRO Run Time System (RTS) detects a problem, diagnostic files are now written to the `\Work\Diagnostics` folder using a naming convention that includes the date and time. The current contents of the black box are also appended to the existing log, providing a trace of the events leading up to the RTS error.

SYSPRO64.DLL exceptions & Client-server failures

A single log file per exception or message is written to the `\Work\Diagnostics` folder instead of all exceptions or messages being concatenated into a single file. This simplifies the SQL Server handling of log files on an individual basis and improves the performance during the upload process. It also lets you check each exception or message as an individual record in the [AdmDiagDetail](#) table.

Micro Focus unhandled exceptions

If an unexpected error occurs in the underlying run time when you log into SYSPRO, the system first verifies whether the `mfdebug.log` file exists in the `\Base` folder before moving it to the `\Work\Diagnostics` folder. The system then creates the new `mfdebug.log` file for the current unhandled exception in the `\Base` folder.

This effectively retains a record of all exceptions instead of just the last example (previously, the original file was overwritten by a subsequent unexpected error).

SYSPRO 8 2020 R2

Operator Settings

As part of the effort to decrease SYSPRO's footprint on the file system, you can enable the new **PERSIST OPERATOR UI SETTINGS TO SQL** setup option (*Setup Options > System Setup > System-Wide Personalization*) to store all operator UI settings files in the [SysSettings](#) table of the system-wide database (usually stored in the `\Base\Settings` folder).

The following file patterns are affected: `*_operator*.*` | `operator.*` | `operator*.*` | `*_operator*.*`

When you enable this option, the process is as follows:

1. When an operator exits SYSPRO, all related operator files located in the `\Base\Settings` folder are compressed and uploaded to the [SysSettings](#) table of the system-wide database.

Operator files are only uploaded when the operator exits SYSPRO.



The `operator.PRF` file is excluded.

2. When an operator logs into SYSPRO, a check is performed to establish whether there are any related operator settings in the [SysSettings](#) table of the system-wide database.

If so, the operator settings are restored from **Microsoft SQL Server** accordingly.

Any files that may already exist on the operator's machine are overwritten with the data from the [SysSettings](#) table of the system-wide database, as the process assumes that the details in the database are the most up-to-date.

3. If SYSPRO terminates abnormally during the current run for the operator, changes made by the operator to the user interface are not saved. When next the operator logs in, the settings from **Microsoft SQL Server** will override existing files on the operator's machine.

This means that the files in the `\Base\Settings` folder remain and are only removed when the operator exits SYSPRO normally.



Enabling the **PERSIST OPERATOR UI SETTINGS TO SQL** setup option indicates that the next time each operator logs into SYSPRO the system will attempt to retrieve operator settings from **Microsoft SQL Server**.

At this stage there won't be any persisted files, so the previous files in the `\Base\Settings` folder are copied to `\Base\Settings\Backup` and the original files in the `\Base\Settings` folder are used as normal.

When the operator logs out of SYSPRO, the files are uploaded to **Microsoft SQL Server** and used from then onwards.

Note that part of this process saves the previous files from `\Base\Settings` to `\Base\Settings\Backup`. The files in the `\Base\Settings` folder are not deleted. The backup is retained while you are using **SYSPRO 8 2020 R2**. Thereafter (i.e. from **SYSPRO 8 2021 R1**) these files will be deleted.

SRS Reports and Documents

From **SYSPRO 8 2020 R2**, enhancements to the SRS architecture cater for the migration of content to the system-wide database:

- Document and report printing
- Report archiving
- Report scheduling
- Report customization

All future custom reports and documents, archived reports and documents and other SRS-related files will be stored in **Microsoft SQL Server**.



All applications that require access to one of these files can then retrieve it using the **COM Retrieve Files Stored on Server** and **Store XML File on Server** business objects.

The custom reports and documents mentioned above are automatically uploaded to **Microsoft SQL Server** during the minor database upgrade to **SYSPRO 8 2020 R2**

In addition, the document archiving architecture has been changed to save the archived documents to the company database when SYSPRO is run in a **SYSPRO Cloud ERP** environment.

The following files are migrated to **Microsoft SQL Server**:

- **SRS Setup Options**

This refers to the file that stores the setup options for **SYSPRO Reporting Services** per company (previously stored in the `\Base\Settings` folder).

The values will be stored on the following system-wide table:

- [SrsSettings](#)

- **Document type custom values and selected archive fields**

This refers to the `DocumentMenu_System.XML` file that was previously stored within the `Base\ReportingCustomized\ReportMenu` folder.

This stores any custom values against a document type, as well as selected archive fields (to be used when archiving documents for the type).

The values will be stored on the following system-wide tables:

- Custom document type details = [SrsDocumentType](#)
- Selected archive field details = [SrsDocArchiveSel](#)

- **Documents**

This refers to all files that were previously stored within the `Base\ReportingCustomized\DocumentControl` folder.

- This stores all the documents created by the customer for all document types. The `Documents.xml` file is used for client-side printing and the `Documents_Server.xml` file is used for server-side printing.

The values will be stored on the following system-wide tables:

- Document details = [SrsDocuments](#)
- Document printer details = [SrsDocPrinters](#)
- Document history details = [SrsDocHistory](#)

■ Document templates

This refers to all document templates that are created for documents. These templates will be stored in the following system-wide table:

- Document templates = [SrsReporting](#)



The **SRS Documents Template Browse** program (accessed from the **Maintain SRS Document Templates¹** and **Add SRS Document Templates²** programs) lets you browse for templates from the database.

■ Report Customization

This includes the following files (previously located in the `\Base\ReportingCustomized` folder):

- Preferences
- ReportControl
- ReportForm
- ReportMenu
- ReportTemplate
- ReportThemes

The customization values will be stored in the following system-wide tables:

- [SrsReporting](#)
- [SrsSettings](#)

■ Report Archives

This includes all files related to the archiving of SRS reports, previously stored in the `\Base\ReportArchive` folder.

¹Program: SRSPM2

²Program: SRSP02

- However, if you defined a custom data folder for your `ReportArchive` in your `IMPACT.INI` file against the `IMPDT1` entry, your archived documents are then normally stored in that location. This folder is then also checked when performing the upgrade.

The values will be stored on the following system-wide table:

- [SrsReportArchive](#)



Only the **SRS Report Archive** files may be deferred during the migration of files to the database, if their cumulative size is larger than 500MB.

■ Report Schedules

This includes all files related to report schedules (previously stored in the `\Base\ReportingCustomized\ReportSchedule` folder).

The values will be stored on the following system-wide table:

- [SrsRptSchedule](#)



The `\BASE\ReportingCustomized\BackUpxxxx-xx-xx` folder (where `xxxx-xx-xx` indicates the date) will contain a backup of all the files that have been imported into **Microsoft SQL Server**.

Deferring the Report Archive Import to SQL



During the migration, the system will determine if your SRS Report Archive files are significant in size (over 500MB) in which case you can choose to defer the migration of these files to allow the rest of the upgrade process to complete and users to start transacting as normal.

Choosing to defer the import means that reports archived prior to **SYSPRO 8 2020 R2** won't be available for viewing until they are migrated. This is performed by running the **SRS Import Archive Files to SQL** program (*Ctrl+R > SRSPAL*).

The time that the import process takes when running the initial database conversion depends on the number of files to process as well as the size of each file. To reduce the time taken for the upgrade and thus allow work in SYSPRO to proceed with minimal disruption, we have provided a facility that enables you to defer the uploading of the **SRS Report Archive files**.

When you run the database upgrade (which occurs on your initial launch of SYSPRO after updating to **SYSPRO 8 2020 R2**), the system checks the size of the **SRS Report Archive folders**.

- *If the size is less than 500MB:*

You are not prompted with the option to defer the import and the files are uploaded as part of the conversion.

- *If the size exceeds 500MB:*

You are provided with the option to import the files immediately or defer to a later stage.

If you choose to defer this file migration, a row is added to the [SrsReportArchive](#) table in the system-wide database to indicate that the archived report migration was deferred.

You can then, at a later stage, use the **SRS Import Archive Files to SQL** program (*Ctrl+R > SRSPAL*) to import the contents of the archive folders into **Microsoft SQL Server**.

When you run the **SRS Import Archive Files to SQL** program, it calls the **SRS Query Functions** business object to upload the files to the database. A tickover is displayed during the process that enables you to cancel the upload at any time. Because the system checks for physical files on the system, the upload process can be restarted as required.

Once all of the files have been uploaded to the database, the row that was added to the [SrsReporting](#) table in the system-wide database is removed.



Reports archived prior to **SYSPRO 8 2020 R2** will not be available for viewing in **SYSPRO Reporting Services** until they have been imported. However, any newly archived reports will be available as they will be archived directly into **Microsoft SQL Server**.

How does this affect users?

Existing reports, menus and forms

Previous customization to reports, forms or menus at a system-wide, company, group, role or operator level, remain intact.



Standard reports, menus, forms, control files and report definition files all remain on the file system.

Creating new reports

You can create a new report in one of two ways:

- From the **New SRS Report** option under the **File** menu of **SYSPRO Reporting Services**, you can use the SYSPRO add-in to save a new report.

By default, this saves the report directly into **Microsoft SQL Server** as `_system` with the relevant control record.

- If you create the report using the standalone version of **SAP Crystal Reports**:

- Save the report file into the `\Base\ReportingCustomized\ReportTemplate` folder.

- Open **SYSPRO Reporting Services**.

- Right-click the folder to which you want to add the report and select the **Add Other Menu Item** option.

The **Add Other Menu Item to xxxx** window is displayed (where 'xxxx' refers to the folder you selected).

- Select **Crystal driver** at the **Report based on** field.
- Enter a name for the report at the **Menu item name** field.
- Select the **Add** option from the toolbar.

The **Report Control** window is displayed.

- Select the browse icon at the **Report template** field.

Locate and select your saved report file.

- Save your changes.

The report is then imported into **Microsoft SQL Server**.



The original report file is moved to the `\BASE\ReportingCustomized\BackUpxxxx-xx-xx` folder.

From this point on, the report is edited through **SYSPRO Reporting Services** as changes are not saved to the file system.

Transferring reports between sites

If a consultant creates a report off-site, you can save the report in **SYSPRO Reporting Services** in one of two ways:

- Use the **Export Report Configuration** tool on the consultant's version of SYSPRO to export the report, form and control file and then import them on the customer's site.
- If the **Export Report Configuration** tool is not available:

- Save the report file into the `\Base\ReportingCustomized\ReportTemplate` folder.

- Open **SYSPRO Reporting Services**.

- Right-click the folder to which you want to add the report and select the **Add Other Menu Item** option.

The **Add Other Menu Item to xxxx** window is displayed (where 'xxxx' refers to the folder you selected).

- Select **Crystal driver** at the **Report based on** field.
- Enter a name for the report at the **Menu item name** field.
- Select the **Add** option from the toolbar.

The **Report Control** window is displayed.

- Select the browse icon at the **Report template** field.

Locate and select your saved report file.

- Save your changes.

The report is then imported into **Microsoft SQL Server**.



The original report file is moved to the `\BASE\ReportingCustomized\BackUpxxxx-xx-xx` folder.

From this point on, the report is edited through **SYSPRO Reporting Services** as changes are not saved to the file system.

Transferring documents between sites

The **Maintain SRS Document Templates**¹ program is used to import and export documents (i.e. you can't drop files into the `DocumentTemplate` folder on the file system).

¹Program: SRSPM2

Editing reports

As custom reports no longer reside on the file system, you must edit/design reports by right-clicking **Report Templates** in *SYSPRO Reporting Services* and then selecting the **Design Report** option from the menu.

Report designers can't edit SRS reports directly through the **SAP Crystal Reports Designer**.

Editing documents

As custom documents no longer reside on the file system, you must use the **Design** option within the **Maintain SRS Document Templates¹** program.

Document designers can't edit SRS documents directly through the **SAP Crystal Reports Designer**.

Adding SRS Document Templates

The user interface of the **Add SRS Document Templates²** program has changed to improve navigation and usability. Options that have changed against the **What would you like to do** field include:

- **Copy sample template**

This is essentially the same as selecting `Copy a template` against the **What would you like to do** field and `Sample template` against the **Source file location** in the old UI prior to *SYSPRO 8 2020 R2*.

- **Copy existing document template**

This is essentially the same as selecting `Copy a template` against the **What would you like to do** field and `Existing template` against the **Source file location** in the old UI prior to *SYSPRO 8 2020 R2*.

- **Use existing document template**


This is essentially the same as selecting `Use an existing template` against the **What would you like to do** field and `Existing template` against the **Source file location** in the old UI prior to *SYSPRO 8 2020 R2*.

- **Import template file**

This is essentially the same as selecting `Copy a template` against the **What would you like to do** field and `External template file` against the **Source file location** in the old UI prior to *SYSPRO 8 2020 R2*.

¹Program: SRSPM2

²Program: SRSP02

-  The import function changes the template name during the import process, so there is no way to bring a template into the system with the same name used when designing it externally.

- **Browse for template file**

This is essentially the same as selecting `Use an existing template` against the **What would you like to do** field and `External template file` against the **Source file location** in the old UI prior to ***SYSPRO 8 2020 R2***.

This displays a list of all files that have been imported into the database, as files no longer reside in the file system.

Maintaining SRS Document Templates

Whereas previously you could browse the file system to select a file that had been manually copied, the browse now functions off the database. You can now only select an existing file that is not already in the system by first using the import function of the **Add SRS Document Templates¹** program.

¹Program: SRSP02



License File

From **SYSPRO 8 2020 R2** the **License Manager** is used to select and import a provided license. The license information resides in the [AdmSystemLicense](#) table of your system-wide database in **Microsoft SQL Server**.

From **SYSPRO 8 2020 R2** all software that requires the license will automatically locate the current license in the system wide database.

If you are using the **SYSPRO Cloud ERP** environment, the license import is applied automatically.

SYSPRO 8 2020 R1

Requirements Planning Snapshot and Scratchpad data

This enhancement optimizes the MRP calculation for SQL without using the file system as a temporary scratch pad.

- Original location: `SNAPSHOT` (user defined)
- SQL Server table(s): `Companydb.Mrp*`
- Mandatory or optional: **Optional**

Affected programs:

- Requirements Calculation

The feature is *disabled* by default so that existing systems work unchanged. To enable the new MRP SQL Optimization, you must uncheck the new processing option: **Run calculation in legacy mode**.

If you are retaining the legacy processing method, then ensure that the snapshot folder still exists.



When using the new MRP SQL Optimization process, the progress bar only displays the text: **Processing Requirements Calculation** during the entire process.

- MRP Requirements Calculation object

The **MRP REQUIREMENTS CALCULATION OBJECT**¹ business object takes advantage of this feature to perform the MRP calculation.



See the [MRP Optimized for SQL Server](#) article for considerations and additional information.

¹Business object: MRPTRC

Related Information

The following related topics can be found within the *Resources > Technical Guides* area of the **SYSPRO Online Help**:

- Database Architecture
- Are there any ISAM files left in SYSPRO 8?
- How does the MRP Requirements Calculation use the temporary scratchpad?



www.syspro.com

Copyright © SYSPRO. All rights reserved.
All brand and product names are trademarks or
registered trademarks of their respective holders.

