

# Database Architecture

## *SYSPRO 8 Technical Article*

Last Published: March 2021



## SYSPRO Help and Reference

Copyright © 2021 SYSPRO Ltd

All rights reserved

No part of this document may be copied, photocopied, or reproduced in any form or by any means without permission in writing from SYSPRO Ltd. SYSPRO is a trademark of SYSPRO Ltd. All other trademarks, service marks, products or services are trademarks or registered trademarks of their respective holders.

SYSPRO Ltd reserves the right to alter the contents of this document without prior notice. While every effort is made to ensure that the contents of this document are correct, no liability whatsoever will be accepted for any errors or omissions.

This document is a copyright work and is protected by local copyright, civil and criminal law and international treaty. This document further contains secret, confidential and proprietary information belonging to SYSPRO Ltd. It is disclosed solely for the purposes of it being used in the context of the licensed use of the SYSPRO Ltd computer software products to which it relates. Such copyright works and information may not be published, disseminated, broadcast, copied or used for any other purpose. This document and all portions thereof included, but without limitation, copyright, trade secret and other intellectual property rights subsisting therein and relating thereto, are and shall at all times remain the sole property of SYSPRO Ltd.

# Contents

---

<b>Introduction</b> .....	<b>5</b>
<b>SYSPRO and Online Transaction Processing</b> .....	<b>6</b>
<b>Supported Versions of Microsoft SQL Server</b> .....	<b>7</b>
SQL Server 2019 .....	7
SQL Server 2017 .....	7
SQL Server 2016 .....	7
SQL Server 2014 .....	7
SQL Server 2012 .....	7
SQL Server 2008 R2.....	7
SQL Server Express .....	8
Service Packs.....	9
How to determine the Version, Edition and Service Packs of your SQL Server?.....	9
SQL Server Compatibility Levels .....	9
<b>SYSPRO 8 Database Architecture</b> .....	<b>11</b>
System-wide Database .....	11
Company Specific Database .....	13
Indexes - Overview.....	22
Foreign Keys.....	24
Stored Procedures - None .....	26
Triggers - None .....	27
Views - None .....	27
Check Constraints - None .....	27
Collation and Case Sensitivity.....	27
Collation and Sorting .....	28
Collation Names .....	29
Using Multiple Collations .....	30
Database Schema .....	31
Other SYSPRO Features that use SQL Server .....	33
Data Stored in the File System .....	33
Static Data Shipped with SYSPRO 8 .....	34

<b>Guidelines: Customizing the SYSPRO Database.....</b>	<b>36</b>
Adhering to the Guidelines and Future Upgrades to the SYSPRO Database .....	36
Ensure that Database User Customization is Necessary .....	37
General Notes about User Customization of the Database .....	38
User Customization - Common Database Objects.....	39
<b>Configuring SYSPRO to work with SQL Server .....</b>	<b>53</b>
Company database authentication .....	54
How does SYSPRO connect to SQL Server?.....	55
Setup Options > System Setup > SQL form reference.....	56
Defining a SQL Server login against each operator ( <i>optional</i> ) .....	60
SQL Connection Strings.....	62
Creating a SYSPRO Database using Management Studio .....	64
What to do when SYSPRO cannot connect to SQL Server.....	68
<b>Insight into SYSPRO Applications and their Interaction with SQL Server .....</b>	<b>71</b>
SYSPRO uses a connection string to connect to SQL Server.....	71
SYSPRO Applications have Two Database Access Methods Available.....	71
SQL Server Profiler.....	74
Transaction Processing .....	75
Optimistic Concurrency Control and Timestamps .....	80
<b>SYSPRO and SQL Server Data Encryption .....</b>	<b>84</b>
Introduction .....	84
Audience.....	84
Why Data Encryption? .....	84
Data Encryption at Rest.....	85
Data Encryption in Motion.....	86
<b>SQL Health Dashboard.....</b>	<b>89</b>
Database Details .....	90
SQL Health Dashboard – Tables .....	93
SQL Health Dashboard – Columns .....	94
SQL Health Dashboard – Indexes .....	95
SQL Health Dashboard – Foreign Keys .....	96
SQL Health Dashboard – Object Dependencies .....	96



SQL Health Dashboard – Index Fragmentation ..... 97

SQL Health Dashboard – SQL Users..... 98

**About the Author ..... 100**



# Introduction

---

SYSPRO 8 is a powerful, scalable and feature rich Enterprise System.

A good understanding of its database architecture, together with how to configure SYSPRO to work effectively with SQL Server, can provide significant benefits to anyone working with SYSPRO.

The key topics covered by this document are:

- [SYSPRO and Online Transaction Processing](#)
- [Supported versions of Microsoft SQL Server](#)
- [The SYSPRO 8 Database Architecture](#)
- [Guidelines for customizing the SYSPRO database](#)
- [Configuring SYSPRO to work with SQL Server](#)
- [Insight into SYSPRO Applications and their Interaction with SQL Server](#)
- [SQL Health Dashboard](#)

This information is aimed at:

- SYSPRO System Administrators
- Database Administrators
- SYSPRO Implementers
- SYSPRO Support Personnel
- SYSPRO Developers
- Anyone else wanting an understanding of SYSPRO's database architecture

# SYSPRO and Online Transaction Processing

---

A significant part of the SYSPRO ERP application is an Online Transaction Processing (OLTP) system and therefore its database design, data access, concurrency, performance, and other issues relate directly to general OLTP guidelines.

The following is an extract from a Microsoft Technical Reference Guide on OLTP systems (<http://technet.microsoft.com/en-us/library/hh393556.aspx>):

*“Operational, or online transaction processing (OLTP), workloads are characterized by small, interactive transactions that generally require sub-second response times. It is common for OLTP systems to have high concurrency requirements, with a read/write ratio ranging from 60/40 to as low as 98/2. Modifications are predominantly singleton statements, and most queries are constrained to simple joins. While limiting joins to as few tables as possible is desirable, a significant number of application systems do join many tables. Standard practices call for indexing strategies in OLTP systems to target an increase in concurrency versus query support; however, more indexes have to be created than is desired to reach acceptable query performance. The lower the proportion of write operations in the system, the higher the level of indexing that can be tolerated, unless the timing of specific write operations is critical. Database plans generally start with third normal form (3NF) enforced with referential integrity (RI) constraints, and then selectively deviate to second normal form (2NF) when necessary to enhance performance.*

*For OLTP systems, the middle tier plays an important role in defining the overall success of the system; while the database is important, it is not the only system component. When a new OLTP system is developed, the initial challenges in scalability and performance are often encountered in the middle tier. It is only after the challenges in the middle tier are addressed that the database tier scalability and performance issues are revealed. We recommend that you refer to the Technical Reference Guides included in the ‘SYSPRO product’ when gathering data points for operational considerations.”*

The only change made to the above extract was that the phrase ‘SYSPRO Product’ was used rather than generic product reference in the original text.

**Note:** Where this document quotes extracts from external resources, they are shown in an *orange* font.

The remainder of this document serves as one of the **SYSPRO Product Technical Reference Guides** mentioned above.



# Supported Versions of Microsoft SQL Server

---

SYSPRO 8 supports data storage using Microsoft SQL Server 2019, 2017, 2016, 2014, 2012, 2008 R2.

It is beyond the scope of this document to cover the various SQL Server editions. However, the vast majority of SYSPRO data functions are fully supported on all versions/editions of SQL Server from SQL Server 2008 R2 onwards.

Always see SYSPRO's Online Help website (<https://help.syspro.com>) for the latest information.

## SQL SERVER 2019

All SYSPRO data functions are available when using Microsoft SQL Server 2019.

## SQL SERVER 2017

All SYSPRO data functions are available when using Microsoft SQL Server 2017.

## SQL SERVER 2016

All SYSPRO data functions are available when using Microsoft SQL Server 2016.

## SQL SERVER 2014

All SYSPRO data functions are available when using Microsoft SQL Server 2014.

## SQL SERVER 2012

All SYSPRO data functions are available when using Microsoft SQL Server 2012.

## SQL SERVER 2008 R2

All SYSPRO data functions are available when using Microsoft SQL Server 2008 R2.

## SQL SERVER VERSIONS – OPTIMIZATION FOR LATER VERSIONS

The majority of SYSPRO code will work equally well with all supported versions of SQL Server listed above.



It should be mentioned that there are some selected functions where application code is further optimized to take account of new capabilities of one or more of the later SQL Server versions. In the case of specific optimization for later SQL Server versions, we still automatically provide alternative code that will work on older versions not supporting the new capabilities. In this case, there may be performance or other optimization benefits when using later SQL Server versions.

## SQL SERVER VERSIONS – ADDITIONAL SYSPRO TECHNOLOGIES

Whilst the core SYSPRO 8 ERP functions support the database versions listed above, some of the newer technologies introduced into recent versions of SYSPRO, have their own database version requirements. Therefore, if you intend to introduce one or more of these technologies, you must consider their platform requirements over and above the core SYSPRO 8 ERP requirements.

These technologies include:

- SYSPRO Social ERP (Harmony)
- SYSPRO Ken the Bot
- SYSPRO IoT
- SYSPRO Connected Services

## SQL SERVER 2008 R2 END OF SUPPORT

The following is a Microsoft statement regarding **SQL Server 2008** – it was copied as of November 2018 (<https://www.microsoft.com/en-gb/sql-server/sql-server-2008>):

### **Prepare for SQL Server 2008 end of support**

*On July 9, 2019, support for SQL Server 2008 and 2008 R2 will end. That means the end of regular security updates. Don't let your infrastructure and applications go unprotected. We're here to help you migrate to current versions for greater security, performance and innovation.*

## SQL SERVER EXPRESS

SYSPRO 8 supports running on Microsoft SQL Server Express (2019, 2017, 2016, 2014, 2012, 2008 R2) – a free version of Microsoft SQL Server. If you intend to use any version of SQL Server Express with SYSPRO, you should make yourself fully aware of the capabilities and limits built-in to the SQL Server Express version you are considering.

These limitations include: *[SQL Server 2012-2019 comments in brackets]*

- Limited database size [10 GB]
- Limited use of server processor sockets and cores [Limited to lesser of 1 Socket or 4 cores]
- Limited access to server memory [Limited to using a maximum of 1 GB RAM]
- Limited tools to help with installation, configuration and database maintenance [Several tools are available including 'SQL Server Management Studio Express']
- Also, there are no Analysis Services, Integration Services or Notification Services

There are many resources covering the various Microsoft SQL Server editions and their capabilities. For example: Features Supported by the Editions of Microsoft SQL Server 2019: <https://www.microsoft.com/en-us/sql-server/sql-server-editions-express>

## SERVICE PACKS

It is good practice to ensure that you review, and where applicable apply, the latest Microsoft Service Packs and any other software patches or upgrades to ensure that you have the best security protection, bug fixes and other enhancements.

## HOW TO DETERMINE THE VERSION, EDITION AND SERVICE PACKS OF YOUR SQL SERVER?

To determine the version and edition, including service packs, of your SQL Server system and components, see the following Microsoft information: <http://support.microsoft.com/kb/321185>

**Note:** The [SYSPRO SQL Health Dashboard](#) provides visibility to some of this version information.

## SQL SERVER COMPATIBILITY LEVELS

When you create a database, it is assigned a compatibility level. Typically, the compatibility level is associated with the current version of SQL Server.

The compatibility level affects the SQL statements that can be issued against the database and the level of query optimization that is used to determine execution plans and ultimately performance.

The compatibility level of a database can be seen using SQL Server Management Studio: Select the database, followed by **Properties > Options > Compatibility level**.

For example:

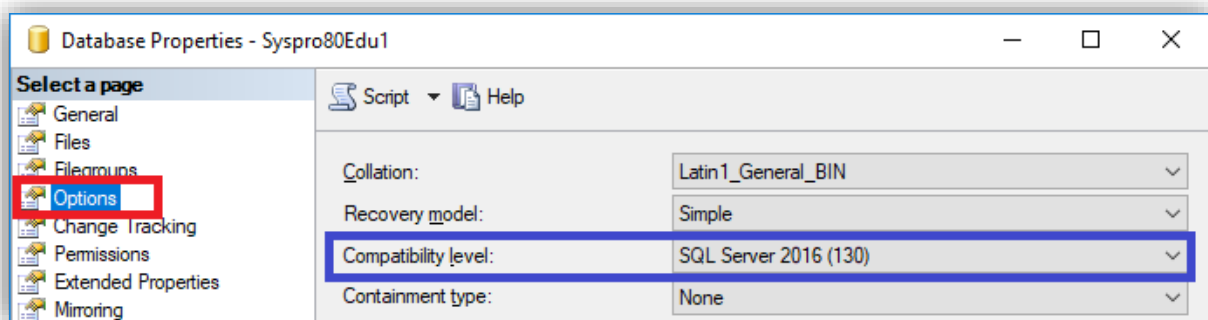


Figure 1: SQL Server Compatibility Level

The following table lists compatibility levels, their associated version of SQL Server and SYSPRO 8 support:

Compatibility level	SQL Server Version	SYSPRO 8 Support
150	SQL Server 2019	Yes
140	SQL Server 2017	Yes
130	SQL Server 2016	Yes
120	SQL Server 2014	Yes
110	SQL Server 2012	Yes
105	SQL Server 2008 R2	Yes
100	SQL Server 2008	No
90	SQL Server 2005	No
80	SQL Server 2000	No

**Note:** You must have a compatibility level of '105' upwards when running SYSPRO 8.

The **SYSPRO SQL Health Dashboard** provides visibility to the database compatibility level. For more information about compatibility levels see the following link: <https://docs.microsoft.com/en-us/sql/t-sql/statements/alter-database-transact-sql-compatibility-level?view=sql-server-2017>

## COMPATIBILITY LEVELS WHEN UPGRADING SQL SERVER

When you upgrade your SQL Server software to a later version, each database's compatibility level remains at its original level number.

For example, if you originally created a SYSPRO 6.0 company using SQL Server 2008 - the company specific database compatibility level would be SQL Server 2008 (100). If you then upgraded your SQL Server installation to SQL Server 2012 the database compatibility level would remain as '100'. If you wished to upgrade to SYSPRO 8 then the compatibility level against the company database would have to be manually upped to SQL Server 2012 (110).

The 'compatibility level' property is just a flag that tells the Database Engine to evaluate SQL syntax and query optimization against this database according to the version of SQL Server specified.

For this reason, it is good practice to update the compatibility level against each SYSPRO database after upgrading your version of SQL Server. It is preferable to make the compatibility level match the version of SQL Server you are using.

Some of our partners and customers have reported that updating the compatibility level from older versions to the most current level can significantly improve the performance of SYSPRO running on SQL Server - this is probably because the later execution plan query optimizer is more efficient.

# SYSPRO 8 Database Architecture

SYSPRO supports multiple companies where each company's data is contained within a single database. In addition, a single system-wide database is used to store information that is not specific to any individual company.

This is easiest to understand using the following examples:

- If a SYSPRO site has a single company then there will be two databases, one for the system-wide data and another for the single company-specific data.
- If the SYSPRO site has two companies then there will be three databases, one for the system-wide data and two more – one each for the two company-specific databases.
- Similarly, if the SYSPRO site has three companies there will be four databases etc.

All SYSPRO databases, both company specific databases and the system-wide database, must be located on a single instance of Microsoft SQL Server.

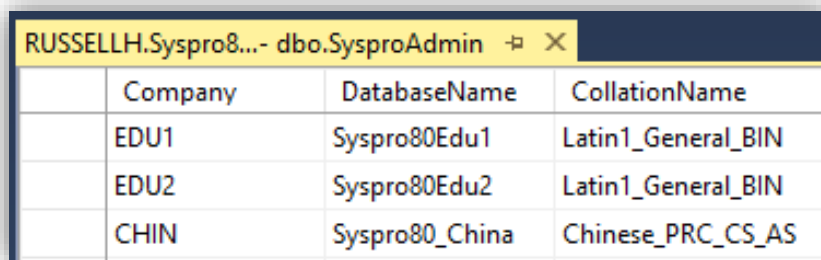
## SYSTEM-WIDE DATABASE

In SYSPRO 8, the system-wide database contains many system-wide tables.

One important table in the system-wide database is named '**SysproAdmin**' – it contains a Company / Database / Collation cross-reference.

See the following example of a '**SysproAdmin**' table in the system-wide database.

In this example there are three SYSPRO companies, the first two have a 'Latin1\_General\_BIN' collation and the third has a 'Chinese\_PRC\_CS\_AS' collation:



Company	DatabaseName	CollationName
EDU1	Syspro80Edu1	Latin1_General_BIN
EDU2	Syspro80Edu2	Latin1_General_BIN
CHIN	Syspro80_China	Chinese_PRC_CS_AS

Figure 2: Example System-wide SysproAdmin Table

The SYSPRO **Setup Options** program (i.e. IMPCFG) allows you to define the name of the system-wide database. You are free to select any name for the system-wide database or retain the default.

See the following screen fragment from the **Setup Options** program:

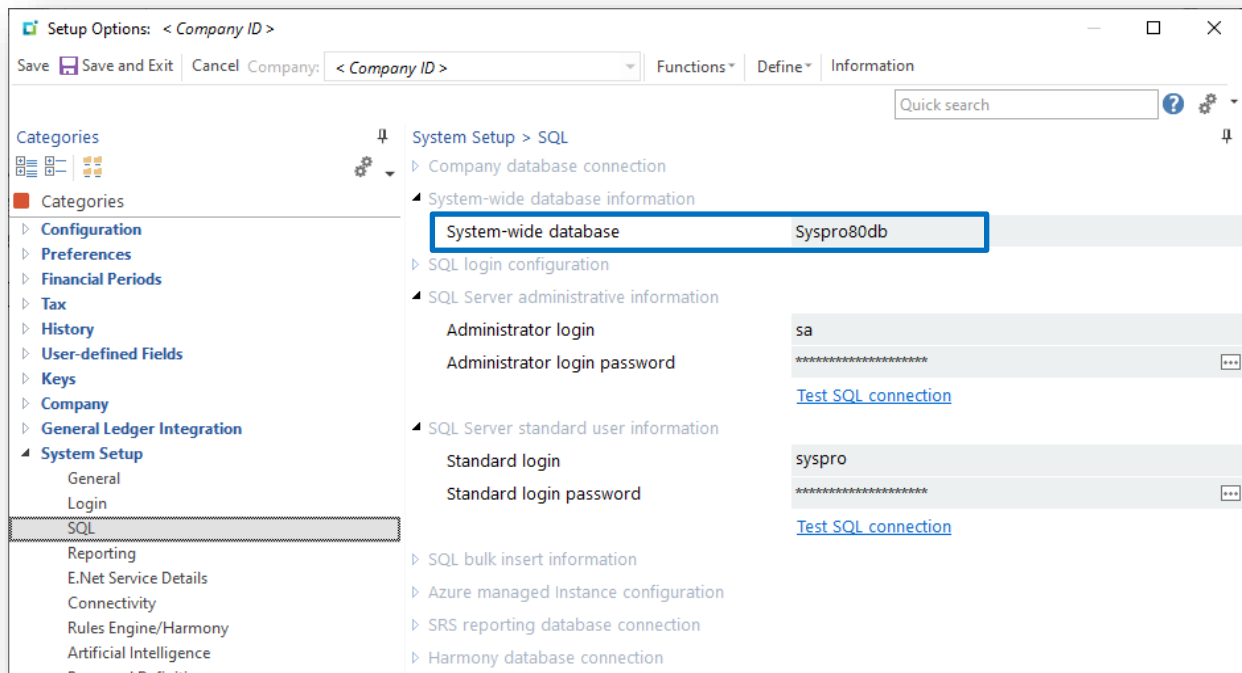


Figure 3: **Setup Options** > System Setup > SQL > System-wide database information

You should not add any additional tables in the system-wide database, nor should you change the **'SysproAdmin'** table structure (columns, data types, indexes, keys, constraints etc.)

The system-wide database name should be a maximum of 18 characters and only consist of:

- Uppercase and lowercase letters (A-Z and a-z)
- Numeric digits (0-9)
- A dash ('-') or an underscore ('\_')

## TYPES OF DATA STORED IN THE SYSTEM-WIDE DATABASE

The system-wide database contains many tables that contain various types of data that are not related to a specific company.

The SYSPRO 8 system-wide database contains over 100 tables depending on the features installed.

These tables contain the following types of data:

- **Company database cross reference**  
This is the 'SysproAdmin' table described above and is used by SYSPRO to determine the relationship between SYSPRO companies and their databases.
- **Data dictionary data**  
This is a set of tables that describe the database structure.  
These tables begin with the characters 'Dds'.

- **Configuration data**

This contains setup and configuration settings such as: Operators; License information; Notepad control; Role, Group and Operator security settings; eSignature configurations; Password control; Database version control and history; Printer configuration.

- **History, Journal and Audit data**

This includes information providing historical journals and audit trails detailing the sequence of messages/transactions, who posted them and when. This includes: System audit log; Operator amendment journals.

- **SYSPRO User information**

Contains information about each active user, the programs they are accessing, the Windows and SQL process IDs, etc.

- **e.net State information**

Contains information about each e.net session.

- **Report Writer information**

The SYSPRO Report Writer database and report definition information is stored in the system-wide database.

These tables begin with the characters 'Rep'.

- **Espresso information**

If you are using SYSPRO Espresso, then a set of tables used for the mobile application is defined in the system-wide database.

These tables begin with the characters 'Esp'.

- **SYSPRO Avanti information**

If you are using the SYSPRO Avanti user interface, then a set of tables used for these applications is defined in the system-wide database.

These tables begin with the characters 'Avt'.

## COMPANY SPECIFIC DATABASE

As mentioned previously, each SYSPRO company has an associated database in SQL Server.

### TYPES OF DATA STORED IN A COMPANY SPECIFIC DATABASE

Each company specific database contains a relatively large number of tables that contain various types of data.

Each SYSPRO 8 company database contains approximately 1000 tables. These tables contain the following types of data:

- **Master data**

This includes information such as Customers, Suppliers, Stock codes, Banks, Prices, Branches and Work centers.

- **Transactional data**

This includes transactional information such as Invoices, Sales orders, Purchase orders, Quotes, Jobs, Bank deposits and withdrawals.

- **History, Journal and Audit data**

This includes information providing historical movements, journals and audit trails detailing the sequence of transactions, who posted them and when.

- **Configuration data**

This contains setup and configuration settings such as the company setup containing: Configuration; Preferences; Financial periods; Tax; History; User-defined fields; Keys; Company and General Ledger Integration.

## STATIC DATABASE STRUCTURE PER SYSPRO RELEASE

New updates to SYSPRO 8 are published periodically (typically twice a year – but this is subject to change). Each release has a static database structure.

This is sometimes called a database schema, but we will call it the database structure as a ‘schema’ has a different meaning in SQL Server and will be discussed as a separate topic later.

By database structure we mean the list of tables, their column names, data types, sizes, defaults, constraints, primary keys, alternate indexes, foreign keys, etc. This excludes the data that resides in each table.

The database structure is static (fixed) for each release of SYSPRO 8. For example, each of the following releases has its own database structure:

- SYSPRO 8 2020 R1
- SYSPRO 8 2020 R2
- SYSPRO 8 2021 R1
- SYSPRO 8 2021 R2

We make every effort not to change any part of the database structure during a software release, however it is possible that we may have to make changes to the database structure during a release in future (for example, for an unexpected legislative change).

## MINOR DATABASE UPDATES

The database changes between SYSPRO 8 releases are called ‘minor database updates’ as they typically consist of new tables and/or new columns and/or new indexes.

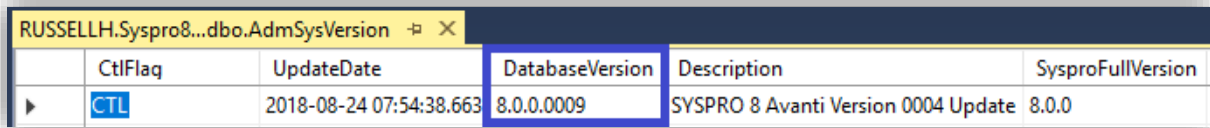
This means that we generally do not remove, rename, or alter existing tables or columns when new SYSPRO 8 releases are made available.

When an administrator logs in to a new SYSPRO 8 release for the first time the minor database update will be applied – this can add new tables, columns, and indexes to both the system-wide database and each company database. All company databases that are linked to the same system-wide database will be updated and must be using the same database version.

## DATABASE VERSION

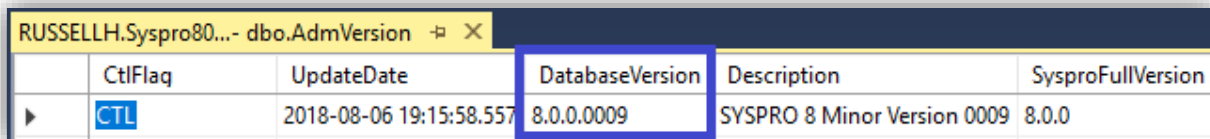
The current system-wide database version can be viewed by querying the **DatabaseVersion** column in the **AdmSysVersion** table.

See the following example:



CtlFlag	UpdateDate	DatabaseVersion	Description	SysproFullVersion
CTL	2018-08-24 07:54:38.663	8.0.0.0009	SYSPRO 8 Avanti Version 0004 Update	8.0.0

Similarly, the current company specific database version can be viewed by querying the **DatabaseVersion** column in the **AdmVersion** table – see the following example:



CtlFlag	UpdateDate	DatabaseVersion	Description	SysproFullVersion
CTL	2018-08-06 19:15:58.557	8.0.0.0009	SYSPRO 8 Minor Version 0009	8.0.0

The **DatabaseVersion** number is not directly related to the naming convention of the SYSPRO 8 release. This is an internal number that is incremented whenever the development team make database changes internally (i.e. these numbers may or may not run concurrently in a production environment).

As of March 2021, the database versions per release were:

SYSPRO Release	DatabaseVersion	Generally Available Release Date
SYSPRO 8 2018 R1	8.0.0.8	July 2018
SYSPRO 8 2018 R2	8.0.0.9	October 2018
SYSPRO 8 2019 R1	8.0.0.10	February 2019
SYSPRO 8 2019 R1	8.0.0.14	August 2019
SYSPRO 8 2020 R1	8.0.0.17	March 2020
SYSPRO 8 2020 R2	8.0.0.21	September 2020
SYSPRO 8 2021 R1	8.0.0.25	April 2021

## DATA DICTIONARY AND DATABASE UPDATES

A set of data dictionary files defining the system-wide and company specific database structures are shipped as part of each SYSPRO 8 release.



When an administrator logs in to a new SYSPRO 8 release for the first time, the minor database update will be applied and the data dictionary (stored as a set of files in the `BASE\DD` folder on the server) is migrated to the system-wide database.

This consists of the following system-wide tables:

Data dictionary files in <code>BASE\DD</code>	Tables in system-wide database	Notes
<b>DDSTAB</b>	DdsTables	Table definitions
<b>DDSCOL</b>	DdsColumns	Column definitions
<b>DDSCVL</b>	DdsColumnValues	Column value text (describes column usage)
<b>DDSIDX</b>	DdsIndexes	Index definition (primary and alternate keys)
<b>DDSFKY</b>	DdsForeignKeys	Foreign key definitions (links between tables)

For this reason, you should treat the contents of these data dictionary tables in the system-wide database as 'read-only', as when the next release is installed, their content will be removed and the new data dictionary information migrated to these tables.

## OPTIONAL TABLES

Although most tables in SYSPRO 8 are static, there are a small set of tables that are optional and may only exist on specific conditions. These include:

- **Custom form data (ending '+')**

SYSPRO 8 provides the ability to store custom data. Each custom form has an associated table in the database. These tables have the same name as the primary associated table and are suffixed with a '+'.  
For example, if you have defined custom form data against a Supplier (primary table name 'ApSupplier') then the custom form data will be stored in a table named 'ApSupplier+'. These tables are stored in the same database as the associated primary table.

For example, if you have defined custom form data against a Supplier (primary table name 'ApSupplier') then the custom form data will be stored in a table named 'ApSupplier+'. These tables are stored in the same database as the associated primary table.

- **Document archive tables (starting 'SrsArchive')**

These tables are prefixed with 'SrsArchive' and are used to provide information about archived documents such as invoices, delivery notes, order acknowledgements, AR statements, factory documentation, etc.

The system administrator can define when to retain these archives (and hence whether the tables exist) and can configure an optional list of attributes that can be used to search the archive (and hence some of the columns are also optional).

- **SYSPRO Espresso**

If you install and configure SYSPRO Espresso (our mobile solution) there are a set of tables beginning 'Esp' that will be created and maintained for Espresso. In this case they are stored in the system-wide database.

As other new technologies are introduced into the SYSPRO ERP product suite, other sets of tables can also be added as and when appropriate.

## TABLE NAMES

SYSPRO table names start with a 2-5-character mnemonic representing the module to which they belong and then a short name describing the table contents. The name uses a camel case style with each 'word' starting with a capital letter followed by one or more lowercase characters.

SYSPRO table names have a maximum of 18 characters.

Table names only consist of uppercase and lowercase letters (A-Z and a-z) and numeric digits (0-9). No other characters are used in standard SYSPRO table names.

Example table names include:

Table name	Table description	Module prefix	Module description
<b>ArCustomer</b>	Customer	Ar	Accounts Receivable
<b>ArInvoice</b>	Customer Invoice	Ar	Accounts Receivable
<b>ApSupplier</b>	Supplier	Ap	Accounts Payable
<b>ApBank</b>	Bank	Ap	Accounts Payable
<b>InvMaster</b>	Inventory Master	Inv	Inventory Control
<b>InvWarehouse</b>	Inventory Warehouse	Inv	Inventory Control

In SYSPRO 8 Custom Form data is stored in separate tables – each table typically has the same name as its primary associated table with a '+' character as a suffix.

For example, custom form data associated with the customer table '**ArCustomer**' is stored in a table named '**ArCustomer+**'.

**Note:** Due to the '+' character at the end of Custom Form table names, you may wish to surround table names with square brackets [ ] when accessing them from third party applications. This avoids the SQL environment treating the '+' character as a concatenation or addition symbol.

## COLUMN NAMES

SYSPRO column names consist of a short name describing the column contents. The name uses a camel case style with one or more 'words' - each 'word' starting with a capital letter followed by one or more lowercase characters.

SYSPRO column names have a maximum of 18 characters.

Column names only consist of uppercase and lowercase letters (A-Z and a-z) and numeric digits (0-9). No other characters are used in standard SYSPRO column names.

Example column names from the Customer table ('ArCustomer') include:

Column name	SQL data type	Column description
<b>Customer</b>	varchar(15)	Customer code
<b>Name</b>	varchar(50)	Customer name
<b>Telephone</b>	varchar(20)	Telephone number
<b>CreditLimit</b>	decimal(12,0)	Credit limit
<b>Salesperson</b>	varchar(20)	Default assigned sales person
<b>Branch</b>	varchar(10)	Default branch
<b>DateLastSale</b>	datetime	Date last sale was made

Each column has several attributes, such as:

- Data type
- Precision, scale, or length
- NULL or not NULL
- Default value
- Collation if data type is CHAR or VARCHAR  
The collation must default to <database default>.

More information about column attributes are described below.

## COLUMN DATA TYPES

SYSPRO columns have one of the following data types:

Data type	Notes	Example data type
<b>char</b>	Character data containing from 1 to 5 characters are stored using the 'char' data type.	char(1) char(5)
<b>varchar</b>	Character data containing 6 or more characters are stored using the 'varchar' data type. The maximum length of a SYSPRO column of type 'varchar' is 3000 characters. Trailing spaces are removed when SYSPRO programs store 'varchar' data.	varchar(6) varchar(15) varchar(255) varchar(1000)

Data type	Notes	Example data type
<b>varchar(max)</b>	Character data containing potentially over 8000 characters. For example, can be used to store XML strings. Trailing spaces are removed.	varchar(max)
<b>varbinary</b>	Binary data can be stored using the 'varbinary' datatype. This is preferred over 'varchar' when the data contains non-readable character data. The maximum length of a SYSPRO column of type 'varbinary' is 3000 characters.	varbinary(60)
<b>varbinary(max)</b>	Binary data containing a potentially large number of characters. For example, can be used to store encrypted strings or an image.	varbinary(max)
<b>decimal</b>	Numeric data containing between 1 and 19 integers and from 0 to 6 decimals is stored in this exact precision numeric data type. In some cases, SYSPRO stores times as an 8-digit numeric field with a format of: HHMMSSFF - Hours, Minutes, Seconds, Fractions of a second (hundredths of a second)	decimal(1,0) decimal(8,0) decimal(12,0) decimal(15,0) decimal(8,5) decimal(14,2) decimal(18,6)
<b>bit</b>	Flags or indicators that can be yes/no or true/false can be stored using the bit datatype. Range: 0 or 1	bit
<b>tinyint</b>	Very small integers can be stored using the tinyint datatype. Range: 0 to 255	tinyint
<b>smallint</b>	Small integers can be stored using the smallint datatype. Range: -32,768 to +32,767	smallint
<b>int</b>	Large integers can be stored using the int datatype. Range: -2,147,483,648 to +2,147,483,647	int
<b>bigint</b>	Very large integers can be stored using the bigint datatype. Range: -9,223,372,036,854,775,808 to +9,223,372,036,854,775,807	bigint

Data type	Notes	Example data type
<b>uniqueidentifier</b>	Globally Unique IDs (known as GUIDs) are stored as type 'uniqueidentifier'. In SYSPRO 8 these are used in various places such as for Contact Management 'ContactId' columns and in the eSignature log for storing GUIDs passed from a 3 <sup>rd</sup> party application.	uniqueidentifier
<b>datetime</b>	All dates are stored using the 'datetime' data type as this is compatible with all supported version of SQL Server. In some cases, SYSPRO stores the time portion in a separate column and only uses the 'datetime' column to store the date part. In this case the time is always stored using an 8-digit decimal field. Also, in this case, you must NOT place a non-zero value into the time part of the datetime column. Alternatively, SYSPRO can store both a date and time in the 'datetime' column. In this case the full precision of the time can be stored – this is to 1/1000 <sup>th</sup> of a second where the last digit is limited to 0, 3 or 7. When not part of the primary key a 'datetime' column can be NULL indicating none or another special value.	datetime
<b>timestamp</b>	This is a database-wide unique number generated by SQL server and used for concurrency control. All SYSPRO tables have a column named 'TimeStamp' with this data type.	timestamp

## NULL VALUES

The majority of SYSPRO 8 columns are defined as 'NOT NULL' – meaning that NULL values are not allowed.

The following list columns are an exception to this general rule - where NULLs are allowed:

- **datetime** - where column is not in a primary key.  
NULL means 'none' or another special value such as 'lowest' or 'highest' depending on context. For example, a stock obsolescence date field could use a NULL date to indicate that an item is not obsolete.
- **uniqueidentifier** – where column is not in a primary key.  
NULL means 'none'.
- **Custom form columns** storing custom form fields that are not part of the primary key.  
NULL means 'none' or 'no value supplied'.

## DEFAULT VALUES

To simplify adding rows to SYSPRO tables, a default constraint is configured against most columns.

This is useful when a third-party application is inserting rows into a table, else you would have to supply a value for each NOT NULL column (which we have seen above is most columns).

The default values are configured 'by data type' – a selection of default values is shown below:

Data type	Default value	Comments
<b>char</b>	Spaces	Default always provided.
<b>varchar</b>	Single space	Default always provided.
<b>varchar(max)</b>	Single space	Default always provided.
<b>decimal</b>	Zero	Default always provided.
<b>datetime</b>	None as NULL is allowed	In cases where a 'datetime' column exists in the primary key, and therefore NULLs are not allowed, a default value of '1900-01-01' is used. It is recommended that you always provide a value to these 'datetime' columns in the primary key.
<b>uniqueidentifier</b>	None as NULL is allowed	In cases where a 'uniqueidentifier' column exists in the primary key, and therefore NULLs are not allowed, a default value of spaces is used. It is recommended that you always provide a value to these 'uniqueidentifier' columns in the primary key.
<b>timestamp</b>	None as value provided by SQL Server	No default provided as this field is updated by SQL Server whenever a row is inserted or changed.

In a SYSPRO 8 database, default constraints are named 'Syspro\_DF\_Table\_Column' where:

- 'Table' is the name of the SYSPRO table,
- 'Column' is the name of the column to which the default is applicable.

For example, the default constraint for the column '**Customer**' in the '**ArCustomer**' table is named:

`Syspro_DF_ArCustomer_Customer`

## INDEXES - OVERVIEW

There are two types of indexes applied to SYSPRO 8 tables:

- Primary Key (Clustered Index) - mandatory
- Alternate Indexes - optional

The following topics describe the difference between primary keys and alternate indexes and discusses performance implications and naming conventions.

### PRIMARY KEY (CLUSTERED INDEX)

All SYSPRO 8 tables have a primary key – usually known as a ‘clustered index’ in SQL Server documentation.

A primary key is one in which no two rows are permitted to have the same index key value (i.e. all values must be unique).

A primary key can consist of one or more columns. When multiple columns are specified, the first column specified is the most important part of the sequence, the second column is the next most important part of the sequence, etc.

All SYSPRO primary keys have all components of the primary key in ascending sequence.

### PRIMARY KEY (CLUSTERED INDEX) - PERFORMANCE

A clustered index is where the logical order of the key values determines the physical order of the corresponding rows in a table. The bottom, or leaf, level of the clustered index contains the actual data rows of the table. A table is allowed one clustered index at a time.

Each SYSPRO table has an appropriate clustered index allowing SYSPRO applications to retrieve a single row with optimal database access.

In addition, applications can retrieve multiple rows, with the best level of performance, by specifying the same sequence as the primary key.

Applications typically do not specify the clustered index name directly but simply specify a condition together with the sequence (ORDER BY) that the rows are to be retrieved. SQL Server determines, using a query optimizer, how to access the data required and will utilize the clustered index when appropriate.

### PRIMARY KEY (CLUSTERED INDEX) - NAMING CONVENTIONS

SYSPRO 8 tables have their clustered keys named by concatenating the table name and the word ‘Key’.

For example, the Customer table ‘**ArCustomer**’ has a clustered key named ‘**ArCustomerKey**’.

## ALTERNATE INDEX

Selected SYSPRO 8 tables have alternate indexes - sometimes also known as 'nonclustered indexes' in SQL Server documentation.

An alternate index on SYSPRO 8 tables is one in which no two rows are permitted to have the same index key value. This is guaranteed by ensuring that alternate indexes always contain all the primary key parts which, as has already been mentioned, are always unique.

The alternate index can consist of just the primary key parts in a different sequence, or it can contain additional columns as well as all the primary key parts.

All SYSPRO alternate indexes always have more than one column - the first column specified is the most important part of the sequence, the second column is the next most important part of the sequence, etc.

All SYSPRO alternate indexes have all components of the alternate index in ascending sequence.

### ALTERNATE INDEX – PERFORMANCE

An alternate index specifies the logical ordering of a table. With an alternate index (nonclustered index), the physical order of the data rows is independent of their indexed order.

As mentioned previously, not all SYSPRO tables have alternate indexes, however when SYSPRO applications need to retrieve rows in a sequence other than the primary key sequence, an alternate index is usually provided as standard when the SYSPRO table was created.

Whilst not always quite as efficient as a clustered index, having an alternate index can provide significant performance benefits when SQL Server uses its query optimizer to determine the most efficient data access method to be used to retrieve data in a sequence other than the primary key sequence.

As for primary keys, applications typically do not specify the alternate index name directly but simply specify a condition together with the sequence that the rows are to be retrieved. SQL Server determines, using a query optimizer, how to access the data required and will utilize an alternate index when appropriate.

If the all the columns to be queried by an application are defined as part of an alternate index the query optimizer may simply retrieve the values from the index instead of the underlying data (this is called a covering index). The query optimizer will always attempt to use the most efficient access method and depending on many factors this may provide additional performance benefits.

### ALTERNATE INDEX - NAMING CONVENTIONS

When a SYSPRO 8 table has an alternate index the alternate index name is generated by concatenating the table name plus the phrase 'Idx', followed by a short mnemonic describing the index using a camel case convention.

As an example, the following lists the primary key and all alternate indexes for the Customer table '**ArCustomer**':



Index name	Type	Description	Key parts
<b>ArCustomerKey</b>	Clustered	Primary key on Customer	Customer
<b>ArCustomerIdxArea</b>	Alternate	Index on Geographic area	Area Customer
<b>ArCustomerIdxBranch</b>	Alternate	Index on Branch	Branch Customer
<b>ArCustomerIdxClass</b>	Alternate	Index on Customer class	Class Customer
<b>ArCustomerIdxEdi</b>	Alternate	Index on EDI sender code	EdiSenderId Customer
<b>ArCustomerIdxFullName</b>	Alternate	Index on Customer name	Name Customer
<b>ArCustomerIdxName</b>	Alternate	Index on Customer short name	ShortName Customer
<b>ArCustomerIdxPhone</b>	Alternate	Index on Telephone	Telephone Customer
<b>ArCustomerIdxSlsp</b>	Alternate	Index on Salesperson	Salesperson Customer

If you are considering adding custom indexes to a SYSPRO table please see the topic: [Guidelines when Adding User Indexes](#).

## FOREIGN KEYS

Foreign Keys allow a database to describe logical relationships between tables.

In SYSPRO 8, a Foreign Key is a link from a single source table to the Primary key of another related table.

For example, the Accounts Receivable Invoice table '**ArInvoice**' has a column that stores the Customer. A Foreign Key between the 'Customer' column in the Invoice table and the 'Customer' column in the Customer table '**ArCustomer**' ensures that this logical relationship is described in the database.

In SYSPRO 8 tables, the number of columns in the Primary key of the target table must match the number of columns in the source table when describing the Foreign Key Relationship.

### FOREIGN KEYS – WITH NOCHECK

In a SYSPRO 8 database, all Foreign Key Relationships are created by specifying 'WITH NOCHECK'.

This means that the actual rows involved in the table relationship are not verified either at the time the table is created or as data is added, deleted, or changed from either the source or target table.

The purpose of Foreign Keys in a SYSPRO database is purely to describe the relationship between tables, not between individual rows of data. The SYSPRO business logic is used to ensure data integrity – not the Foreign Key Relationships described in the database.

## FOREIGN KEYS – NAMING CONVENTIONS

When Foreign Keys are described in a SYSPRO 8 database, the name of the Foreign Key is 'Syspro\_FK\_' followed by the source table and then an underscore and then the target table.

For example, the Foreign Key Relationship between the '**ArInvoice**' table and the '**ArCustomer**' table is named:

```
Syspro_FK_ArInvoice_ArCustomer
```

## FOREIGN KEYS – DATABASE DIAGRAMS (ENTITY RELATIONSHIP DIAGRAMS)

The definition of Foreign Keys in the SYSPRO database enables the use of tools such as the SQL Server Diagram tool. These tools use Foreign Key Relationships as described in the database to understand the relationships between tables. This is more accurate than comparing column names for the relationships - as used in some database visualization tools.

It's relatively simple to create a new Diagram in SQL Server by selecting the Database > 'New Database Diagram' and then selecting two or more tables (e.g. 'ArCustomer' and 'ArInvoice') and viewing the resultant diagram. (It can take a few minutes to zoom in/out and re-organize the tables shown to best view the generated diagram – especially when there are many tables)

See the following example:

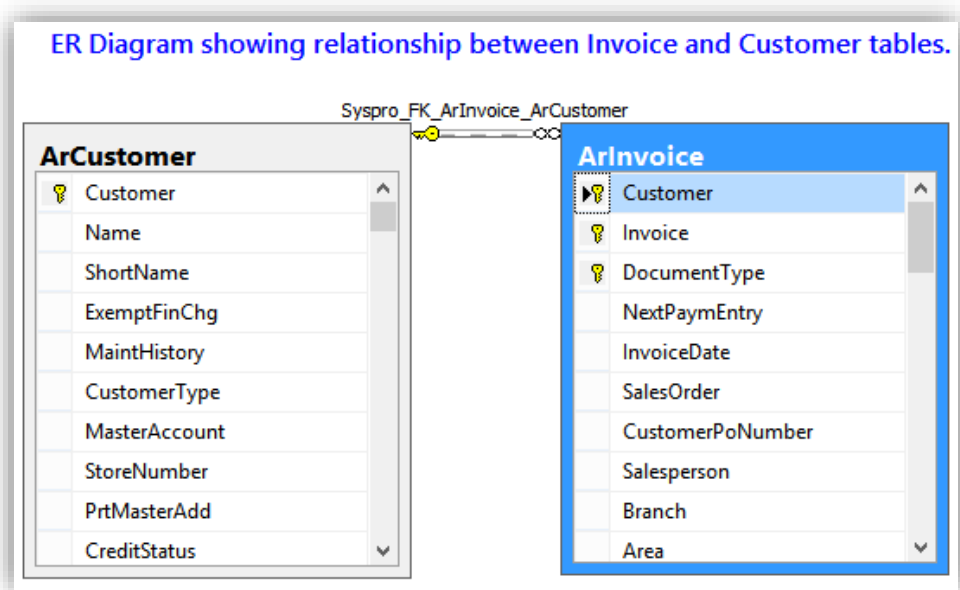



Figure 4: Simple Entity Relationship Diagram



When you exit after creating a new database diagram you will be prompted to give the diagram a name – this can be used to later retrieve and show and/or edit the diagram.

This diagram is interactive – i.e. you can use the context menu to view the relationships between tables, view keys, view check constraints and view column properties.

**Warning:** Take care when using the SQL Server Diagram tool to ensure that you only change the diagram and view the database structure, and not change the database in any way.

This is easiest accomplished by ensuring you do not access the Diagram tool when logged in as a system administrator.

## STORED PROCEDURES - NONE

SYSPRO does not use Stored Procedures in its database.


Stored Procedures are typically created by applications for one or more of the following reasons:

- They are used to provide standard functions with documented parameters that the application calls to perform a query or change the database. They can contain simple SQL statements or contain more sophisticated business logic. By documenting available Stored Procedures and their parameters, third party developers are also able to take advantage of the standard functions provided.
- They are created to improve the performance of an application when it needs to make repetitive calls to the database engine.
- They are created to improve the performance of an application when many SQL statements are to be 'batched together' in a single transaction as the code is executed on the SQL Server and does require a roundtrip to the application server.

In SYSPRO, the first reason for having a Stored Procedure in a database (i.e. standardized and documented functions) is handled by SYSPRO Business Objects. A Business Object provides a simple, documented, and extensible API layer into SYSPRO's business logic. Instead of placing business logic in a Stored Procedure, SYSPRO places its business logic inside its own Business Objects.

The second reason for having Stored Procedures (i.e. performance) is handled by SYSPRO taking advantage of SQL Server's execution plan caching and reuse. When virtually identical SQL Queries are issued to SQL Server, it generates an execution plan based on its query optimizer. This parameterized execution plan is cached in the 'execution plan' cache. SYSPRO programs are carefully developed to ensure that where relevant the same SQL parameterized statement is issued so that the performance benefit of not having to generate a new execution plan (but use one already cached) is taken advantage of.

This topic is explored further: [http://technet.microsoft.com/en-us/library/ms181055\(v=sql.105\).aspx](http://technet.microsoft.com/en-us/library/ms181055(v=sql.105).aspx).



There are other reasons why some applications use Stored Procedures, such as by avoiding roundtrips. However, by making available appropriately coded SYSPRO Business Objects the level of performance, scalability, programmability, and extensibility of the interface in most cases more than makes up for any lack of Stored Procedures.

## TRIGGERS - NONE

SYSPRO does not use SQL Triggers in its database. End users can develop conditional actions in many ways in the SYSPRO software by making use of the following technologies:

- **eSignatures**
  - eSignatures can be configured to fire a trigger when a transaction occurs with a user-defined condition.
  - eSignatures can be configured by operator, role, group, company or system-wide.
  - eSignature triggers can call SYSPRO applications, call third party applications, run a VBScript and invoke a Business Object, send a notification email, etc.
- **Events and Triggers**

These two technologies also provide the ability to call applications, send email, etc.
- **Workflow**

This is a full-blown workflow engine that can provide very sophisticated abilities to detect transaction success or failure, prompt for authorization or approval, trigger virtually any sort of application or notification based on completely user-defined conditions.

## VIEWS - NONE

SYSPRO does not create, maintain, or use any SQL Server Views in the SYSPRO database. In SYSPRO applications, all database access is directly with base tables.

## CHECK CONSTRAINTS - NONE

SYSPRO does not create, maintain, or rely on Check Constraints. All business logic is encapsulated in the various SYSPRO Business Objects.

## COLLATION AND CASE SENSITIVITY

SYSPRO requires that all its databases have a case sensitive collation. This includes collations known as 'Binary'.

The majority of SYSPRO 8 character data is stored in columns with a data type of CHAR or VARCHAR and these columns are affected by the collation when comparing data or sequencing results.

When you add a SYSPRO 8 database you should ensure that the database collation is either 'Binary' or 'Case Sensitive'.

See the following examples:

Collation name	Notes
<b>Latin1_General_BIN</b>	A Binary collation – providing high performance and covering English and many Western European languages. Used for backward compatibility between previous Binary collations.
<b>Latin1_General_BIN2</b>	A Binary collation – providing high performance and covering English and many Western European languages. A newer Binary collation available in more recent versions of SQL Server. It is recommended that the 'BIN2' collation is used rather than a 'BIN' collation – however if you have already created a database with a 'BIN' collation you do not need to consider changing the existing collation.
<b>Latin1_General_CS_AS</b>	A Case Sensitive and Accent Sensitive collation – covering English and many Western European languages.
<b>Chinese_PRC_CS_AS</b>	A Case Sensitive and Accent Sensitive collation – covers the simplified Chinese language used in mainland China and Chinese speaking territories. Can also store English data.

## COLLATION AND SORTING

The Collation not only affects comparison between items but also the sort order when retrieving items. The type of collation affects the sequence by which items are retrieved:

- **Binary collations**  
Typically returns all uppercase items before all lowercase ones.
- **Case Sensitive collations**  
Typically returns each item alphabetically regardless of case (within the same letter lowercase shows before uppercase).

To demonstrate this - assume you have the following items:

aa, AA, bb, BB, cc, CC, cat, Cat, dog, Dog

If the application selects all rows ordering them in ascending sequence, then the items will be returned as shown in the following table depending on the collation:

Latin1_General_BIN	Latin1_General_CS_AS
AA	aa
BB	AA
CC	bb
Cat	BB
Dog	cat
aa	Cat
bb	cc
cat	CC
cc	dog
dog	Dog

From this you may see that the 'Case Sensitive' collations are often preferable (and less confusing to operators) than the 'Binary' collations. However, 'Binary' collations can offer the best absolute performance.

## COLLATION NAMES

There are several thousand available collations in SQL Server depending on the SQL Server version.

There are many variations to the style of collation names – in addition to the ones mentioned above, some start 'SQL\_' and some contain digits such as '90' or '100' in them.

You can select a required collation as long they are either 'Binary' or Case Sensitive and Accent Sensitive. If in doubt, contact your SYSPRO support team for advice.

**Important:** You must not configure SYSPRO to run on a Case Insensitive collation else you may experience unexpected results. For similar reasons we suggest not using an Accent Insensitive collation.

The letters 'CI' in the collation name mean 'Case Insensitive' and 'AI' means 'Accent Insensitive'. Therefore, avoid collations with 'CI' or 'AI' and rather use a collation that includes the letters 'CS\_AS' (or 'BIN' or 'BIN2' for Binary).

## USING MULTIPLE COLLATIONS

The majority of SYSPRO sites have the same collation against all SYSPRO databases, whether they are the system-wide database or one of the company specific databases.

However, SYSPRO 8 does support one or more company databases having different collations from each other.

An earlier topic discussed the system-wide database and the '**SysproAdmin**' table. If you recall the '**SysproAdmin**' table contains a single row, one for each company. It not only contains the company id and database name, but it also contains the database collation. This is the mechanism by which SYSPRO 8 companies can have different collations.

### MULTIPLE COLLATIONS – SINGLE COLLATION PER DATABASE

In a single SYSPRO database all columns containing character data must have the same collation.

**Note:** SYSPRO does not support some columns having one collation whilst other columns in the same database have a different collation.

If you have two or more company databases, then each can have a different collation. If you have Shared GL or Shared Inventory, see the following topics:

### MULTIPLE COLLATIONS – SHARED GL SUPPORTED

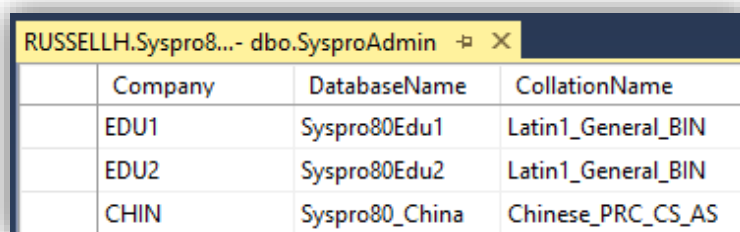
The primary purpose of using Shared GL is to provide consolidated reporting across multiple companies.

For example, using the Financial Report Writer, it's possible to report on all GL data for a specific branch, department, or account code across two or more companies. This can only be performed when the companies to be consolidated have the same Shared GL id.


The concept of Shared GL is that you can have two or more SYSPRO companies each storing their main General Ledger data in one of the company's databases. Each row in each of the Shared GL tables in the shared database have a company id in the key ensuring that all data is kept separately 'per company'. They just happen to be stored in a single database.

SYSPRO 8 does support two or more companies each having different collations but having a single Shared GL id.

To clarify further, consider the following example: Assume your site contains the three companies as defined in the following '**SysproAdmin**' table:



Company	DatabaseName	CollationName
EDU1	Syspro80Edu1	Latin1_General_BIN
EDU2	Syspro80Edu2	Latin1_General_BIN
CHIN	Syspro80_China	Chinese_PRC_CS_AS



Also assume that SYSPRO companies 'EDU1' and 'EDU2' have been defined as having a Shared GL id of 'CHIN', all General Ledger data will be stored in the 'CHIN' company's database named '**Syspro80\_China**'. Therefore company 'EDU1' (database '**Syspro80Edu1**') will have most of its character data stored using the 'Latin1\_General\_BIN' collation.

Similarly, company 'EDU2' (database '**Syspro80Edu2**') will have most of its character data also stored using the 'Latin1\_General\_BIN' collation.

Company 'CHIN' (database '**Syspro80\_China**') will have all character data stored using the 'Chinese\_PRC\_CS\_AS' collation.

However, both companies 'EDU1' and 'EDU2' General Ledger data will be stored in company 'CHIN' (database '**Syspro80\_China**') and this will be stored using the 'Chinese\_PRC\_CS\_AS' collation.

Remember that company 'EDU1' General Ledger data is stored with a company id of 'EDU1' recorded as a key item against each row in each of the General Ledger tables and therefore are separate from company 'EDU2' and 'CHIN'. Even though they are all stored in a single set of GL tables.

For the consolidated reporting purposes, you would also typically have consistency of GL code naming or numbering – often using numeric digits for your GL codes.

## MULTIPLE COLLATIONS – SHARED INVENTORY [NOT SUPPORTED]

SYSPRO 8 does not support Shared Inventory between companies that have databases of different collations.

Using the example '**SysproAdmin**' above – Company 'EDU1' could have a Shared Inventory id 'EDU2' as they have the same collation.

However, company 'EDU1' could not have a Shared Inventory id of 'CHIN' as these have different collations.

The difference between Shared GL and Shared Inventory is that Shared GL stores the items from each company separately in the target database (by storing the source company id in each row) whereas Shared Inventory is using the same rows to store items (Inventory Master) and quantities (Inventory Warehouse, Bins, Lots, Serial's etc.)

## DATABASE SCHEMA

SQL Server has a concept of a database schema. This topic describes what this means, how it is typically used in SQL Server, and how SYSPRO should be configured and will reference the schema.

### WHAT IS A DATABASE SCHEMA?

A Database Schema in SQL Server is a container of objects within a database.



Databases have a default schema named 'dbo' (database owner). In most cases when objects are created, they are created under the 'dbo' schema.

It is possible to create other schemas against a database and then create objects in these schemas (instead of the 'dbo' schema). As the schema forms part of a full qualified object name (see below), you can have objects with the same name in different schema – i.e. the schema forms part of the uniqueness when naming objects.

The primary purpose of schemas is to simplify applying permissions to objects in SQL Server as it is relatively simple to assign a default schema to a SQL login and ensure that the login only has access (read, update, full, etc.) against objects in that schema.

For this reason, it's often not recommended that you have objects with the same name in different schema. It can be confusing to all database users and developers.

## FULLY QUALIFIED OBJECT NAMES

All database objects can be uniquely identified by using a four part fully qualified object name. The parts consist of:

1. Server name
2. Database name
3. Schema name
4. Object name

For example, the AR Customer table '**ArCustomer**' stored on a server named 'SysproServer' in a database named '**SysproCompanyA**' would be fully qualified as:

```
[SysproServer] . [SysproCompanyA] . [dbo] . [ArCustomer]
```

The third part of the object definition is the schema name - in this case 'dbo'. In this example all parts of the name have been surrounded by brackets [ ]. This is optional when specifying objects unless there are special characters in the object name such as a space or '+'.

## SYSPRO AND THE 'DBO' SCHEMA

Most SYSPRO applications do not explicitly reference the schema when accessing database objects. In this case, SQL Server applies a default schema name inherited from the SQL login being used. By default, most SQL logins have a default schema name of 'dbo'.

For this reason, all SYSPRO objects must reside in a single schema named 'dbo'. And all SQL logins used by the SYSPRO application must have a default schema of 'dbo'.

If you wish to create your own schema with you own objects, see the [User Customization](#) topics later in this document. However, standard SYSPRO objects must all reside in a single schema named 'dbo'.

## OTHER SYSPRO FEATURES THAT USE SQL SERVER

When installed and configured appropriately the following SYSPRO modules/technologies also store information in SQL Server - this includes:

- **SYSPRO Workflow Services**

The SYSPRO Workflow Services (SWS) module creates its own database and tables that are separate from the database architecture explained in this document. They are not explained any further here.

- **SYSPRO Reporting Services**

SYSPRO 8 allows SYSPRO Reporting Services (SRS) to use SQL Server as a temporary storage mechanism when producing reports.

See the **Reporting** form in the **Setup Options** program (i.e. IMPCFG) for more information (*Setup options > System Setup > Reporting*).

These temporary tables are not explained further here.

## DATA STORED IN THE FILE SYSTEM

A relatively small amount of configuration and setup data is stored in the file system. These files are not stored in the system-wide or company specific databases.

You should be aware when performing maintenance task that these files require regular backup.

Note that 'no transactional data' is stored in the file system – only configuration, setup, and preference files. This means that full database transactional integrity will be maintained in the event of a system or other failure.

You will note that since SYSPRO 8 2020 R2, we have added options for most of these to be persisted to SQL Server if you wish.

The information stored in the file system includes:

- **System configuration file**

The `IMPACT.INI` file contains some system-wide licensing information and how to connect to SQL Server – this file is stored in the `WORK` folder.

- **Preference settings**

Preferences and saved settings as users run applications can be persisted to files stored in the `BASE\SETTINGS` folder.

From SYSPRO 8 2020 R2 onwards there is a system-wide preference allowing the preferences and settings to be persisted to SQL Server.

- **Custom reports**

When designing custom reports using the SRS technology, they are stored in a folder named:

- `BASE\ReportingCustomized`

From SYSPRO 8 2020 R2 onwards, Custom reports are always persisted to SQL Server. Therefore, during the upgrade process you can elect to either upload the customized reports immediately or continue the upgrade and upload them later. However, all customized reports must be uploaded to SQL Server before they can be edited or run.

- **Rich text notes**

Most notes in SYSPRO are stored as Rich Text (RTF) files. These are stored in the file system under an appropriate folder, such as:

- `WORK\Notes`
- `WORK\crm_XXXX\activity\body` (*where XXXX is the company*)

From SYSPRO 8 2021 R1 onwards, there is a system-wide setting allowing the preferences and settings to be persisted to SQL Server.

Run the **Migrate Files To SQL** program (i.e. **IMPCFL**) to initiate the process.

- **Contact Management activity attachments**

When posting or recording an Activity in Contact Management (such as receiving an email), there can be multiple attachments.

Each attachment is stored with an appropriate name in a folder:

- `WORK\crm_XXXX\activity\attach` (*where XXXX is the company*)

From SYSPRO 8 2021 R1 onwards, there is a system-wide setting allowing the Contact management activity body text and attachments to be persisted to SQL Server.

Run the **Migrate Files To SQL** program (**IMPCFL**) to initiate the process.

## STATIC DATA SHIPPED WITH SYSPRO 8

A selection of files is shipped as part of the SYSPRO 8 application providing static lists, messages and other information used by the SYSPRO 8 applications.

These files are stored in the file system and do not require backup as they are installed as part of the SYSPRO application.

In some cases, SYSPRO may upload selected files to SQL Server. In this case, the SQL data must always be treated as read-only as the system may refresh this data when new versions of the files are loaded.



These include:

- **SYSPRO 8 data dictionary**

This is a set of files that describe the database structure used by SYSPRO 8.

Location: `BASE\DD`.

Remember that parts of the data dictionary are uploaded to SQL Server each time a new SYSPRO release is installed.

- **Business object schema and sample XML**

This is a set of schema and sample XML input and output for each business object available for e.net and VBScript developers.

Location: `BASE\SCHEMAS`.

- **Various lists of items**

These are several plain text files (typically ending with `.IMP`) that contain information such as a list of all the:

- SYSPRO programs
- business objects
- available modules,
- available custom forms
- configurable documents.

Location: `BASE\Store`.

# Guidelines: Customizing the SYSPRO Database

---

This set of topics provides guidelines for a database administrator, software implementer, SYSPRO support personnel, or anyone who is considering modifying the SYSPRO database in any way.

## ADHERING TO THE GUIDELINES AND FUTURE UPGRADES TO THE SYSPRO DATABASE

If you strictly adhere to all the guidelines documented in this Database Customization topic, then when you upgrade to a later version of SYSPRO (requiring an upgrade to the SYSPRO database) or a later release of SYSPRO 8 (requiring a minor database update) the conversion process will take account of your User Customization.

In some circumstances this may include:

- Saving and then dropping your User Customization
- Performing the database upgrade
- Reapplying your User Customization upon completion

Although not common, in some cases the upgrade process may rename some existing columns, or some columns may be relocated to different tables. You will be notified if any User Customization could not be re-applied due to these reasons.

Remember that the User Customization was saved prior to being dropped, so it is typically a relatively simple task to review the reason for the failure and then to manually re-apply the User Customization after making the necessary changes.

Typically, these types of issues are the only situations in which your User Customization will not be retained during a database upgrade.

To help you plan for an upgrade, we typically supply a 'readiness program' in the prior version of SYSPRO. This not only performs a health-check of your current database, but it also allows you to verify that any User Customization will be handled during the upgrade process.

**Note:** The 'minor database upgrade' that is performed when updating from one release of SYSPRO 8 to another (such as SYSPRO 8 2018 R1 to SYSPRO 8 2018 R2), typically applies only new tables, columns and indexes and therefore, (as long as you adhere to the guidelines in this topic) your User Customization will not be affected during the update process.

## ENSURE THAT DATABASE USER CUSTOMIZATION IS NECESSARY

Before considering any modification or addition to a SYSPRO database, we strongly suggest that you make yourself aware of the rich set of customization capabilities built-in to the SYSPRO application. There are many types of configuration options, settings and parameters and other customizations, together with 'power tailoring' capabilities configurable at the system-wide, company, role, operator and/or other levels.

All these standard SYSPRO functions provide futureproofing as any customization will be handled in future versions and releases of software. Some of these are listed below:

### BUSINESS OBJECTS AND THE SYSPRO APPLICATION

The SYSPRO application provides a rich set of Business Objects with an open and extensible application interface (API) accessible from any language.

These Business Objects encapsulate the SYSPRO business logic ensuring consistent application of configuration options, settings, and other rules.

Wherever possible you should use the SYSPRO User Interface or use Business Objects to read, change or delete SYSPRO data or to post any transaction.

### POWER TAILORING – THE SYSPRO USER INTERFACE

The SYSPRO user interface has a highly customizable 'power tailoring' capability.

You can configure the user interface from a cosmetic point of view or fully customize the user experience including using powerful VBScripts.

These VBScripts can be used to test and limit or highlight entry of values in single or multiple fields and lookup values against user tables. Messages can be shown to the operator when required.

You can also invoke Business Objects to lookup, validate or post values as required.


It is recommended that you use the 'power tailoring' capabilities rather than adding a User Check Constraint or User Trigger to the database.

### E-SIGNATURES, EVENTS AND TRIGGERS

SYSPRO provides a rich set of functions to fire triggers when transactions or other program specific functions occur.

These triggers can include:

- Sending email notifications
- Running applications
- Executing VBScripts
- Running pre-defined reports
- Logging information in summary and detail logs
- Creating a new entry in a message inbox



When using eSignatures you can provide additional authentication and even prevent transactions before they occur.

## CUSTOM FORMS

The Custom Form system allows you to configure one or more user defined values against key fields.

For example, you could record three user defined fields related to your specific business rules whenever a sales order is captured. Each field can have a name, caption, data type, size and default specified together with various validation rules.

Custom Form fields are not only visible in Entry and Display Forms but can be included in list views, reports, printed on documents and used by the Report Writer.

Third party applications can add, change, delete and retrieve Custom Form fields using relevant Business Objects avoiding direct access to the SYSPRO database. All validation rules are honored.

Many SYSPRO sites use Custom Forms – often avoiding the requirement to add User Columns or even User Tables to the SYSPRO database.

## GENERAL NOTES ABOUT USER CUSTOMIZATION OF THE DATABASE

You should only make changes to the SYSPRO database when necessary. Preferably after a full and detailed business case has been agreed by all stakeholders.

As a rule:

- We do not recommend that the SYSPRO database is accessed directly by any application.
- We do not recommend that the SYSPRO database is modified or enhanced in any way.

However, there are some business cases where modifying or adding to the standard SYSPRO database is the most practical way of providing the required business functionality.

The remainder of this 'User Customization' topic is designed to provide some guidelines to ensure that any User changes to the SYSPRO database cause the minimum of overhead or other problems to the SYSPRO application. Just as importantly, these guidelines help to ensure that any changes or additions made will have minimal effect with future upgrades to the SYSPRO application and its database.

Although this information is provided as a set of guidelines - any additional customization not explicitly approved here could lead to unexpected (or even intermittent behavior) by one or more parts of the SYSPRO application and is therefore not recommended. We have had many cases in the past where inappropriate database customization has caused poor performance, caused transactions to fail or other unexpected or unwanted behavior.

## USER CUSTOMIZATION – NAMING CONVENTIONS

Generally, any User Customization to a SYSPRO database should consider the fact that future upgrades to the SYSPRO database may add new objects that could conflict with your User Customization.

Each topic below, where user objects can be created, describes the naming conventions used in the standard SYSPRO database. One simple suggestion to avoid conflict with SYSPRO objects is to prefix any user objects with the string 'Usr\_'.

For example, User Tables could be named 'Usr\_Desks', 'Usr\_Chairs' and 'Usr\_Tables'. Similarly, User Columns could be named 'Usr\_Count' and 'Usr\_Notation'.

Instead of the prefix 'Usr\_' you might consider using your company or product name together with an underscore. The reason for suggesting an underscore character is that it is a character not found in most objects (such as tables, indexes, or column names) in a standard SYSPRO database.

If you intend to access your User Tables and User Columns from the SYSPRO Report Writer, or from some generic Business Objects, then we recommend that you keep the Table and Column names to a maximum of 18 characters. The Report Writer and SYSPRO data dictionary limit these object names to 18 characters.

## USER CUSTOMIZATION – DOCUMENTATION

As all User Customization should only be performed once a business case for the customization has been documented and approved by relevant stakeholders, it is important to create and retain a document covering all User Customization of the SYSPRO databases together with the usage of these objects.

This should include a list of all applications that access the User Objects.

By authoring and maintaining this documentation, administrators, developers, and support personnel can quickly understand the extent of any customization to the standard SYSPRO database and the reasons for the changes.

## USER CUSTOMIZATION - COMMON DATABASE OBJECTS

The following topics discuss the most common database objects and our guidelines as far as User Customization is concerned.

### DROPPING OR RENAMING OF STANDARD TABLES – NOT ALLOWED

You must not drop or rename a standard SYSPRO table.

### DROPPING OR RENAMING OF STANDARD COLUMNS – NOT ALLOWED

You must not drop or rename a standard SYSPRO column.



You also must not make any changes to the standard SYSPRO column attributes or properties. This includes:

- NULL and NOT NULL indicator
- Data type
- Length, precision, or scale
- Default value (if any)
- Check constraints

## DROPPING OR RENAMING OF STANDARD KEYS/INDEXES – NOT ALLOWED

You must not drop, rename, or change in any way a standard SYSPRO primary key or alternate index. This includes:

- Changing the columns that make up the key or index
- Changing the Ascending/Descending sequence of any key or index

## USER TABLES - ALLOWED

User Tables describes the requirement to add additional tables, over and above standard SYSPRO tables.

Generally adding User Tables has no effect on the running of the SYSPRO application. However, this assumes that you have considered the following:

- The naming convention of any User Tables must not conflict with current or possible future SYSPRO table names. You could achieve this by reviewing the naming conventions for standard SYSPRO tables:
  - SYSPRO table names start with a mnemonic of one of the SYSPRO modules and then consist of a set of one or more words using a camel case style with initial capital letters.
  - SYSPRO table names only consist of uppercase and lowercase letter (A-Z and a-z) and digits (0-9).
  - Exceptions include custom form tables as they can be suffixed with a '+'.
    - In addition of our Harmony tracking tables can be suffixed by a '-'.

## GUIDELINES WHEN ADDING USER TABLES

Due to the above, we recommend that any User Tables should have a different naming convention from the standard SYSPRO tables.

A suggestion could be that User Tables are prefixed with 'Usr\_' followed by the name of the table.

By using a suitable naming convention (such as that described above) you can avoid possible future conflict with SYSPRO table names.

If you intend to access your User Table in the SYSPRO Report Writer or other standard SYSPRO applications, then you should ensure that the table name is no longer than 18 characters.

## USER COLUMNS - ALLOWED

User Columns in SYSPRO tables describes the requirement to add additional columns, over and above standard SYSPRO columns, to a standard SYSPRO table.

Generally adding User Columns has no effect on the running of the SYSPRO application. However, this assumes that you have considered the following:

- The naming convention of any User Columns must not conflict with current or possible future SYSPRO column names.

You could achieve this by reviewing the naming conventions for standard SYSPRO columns:

- SYSPRO column names consist of a short name describing the column contents. The name uses a camel case style with one or more 'words' - each 'word' starting with a capital letter followed by one or more lowercase characters.
  - SYSPRO column names have a maximum of 18 characters.
  - Column names only consist of uppercase and lowercase letters (A-Z and a-z) and numeric digits (0-9). No other characters are used in standard SYSPRO column names.
- User Columns must be configured in such a way so that there is no effect on regular SYSPRO applications inserting, updating, or deleting rows in any standard SYSPRO table.
  - You should not add columns with the Identity property.

*If you require to add a computed column, see the [User Computed Columns - Allowed](#) topic.*

## GUIDELINES WHEN ADDING USER COLUMNS

Due to the above, we recommend that any User Columns should have a different naming convention from the standard SYSPRO columns.

A suggestion could be that User Columns are prefixed with 'Usr\_' followed by the name of the column.

By using a suitable naming convention (such as that described above) you can avoid possible future conflict with SYSPRO columns names.

If you intend to access your User Column in the SYSPRO Report Writer or other standard SYSPRO application, then you should ensure that the column name is no longer than 18 characters.

In addition to the above naming convention issues, you should also be aware of the following before adding User Columns:

- User Columns must be configured in such a way so that there is no effect on regular SYSPRO applications inserting, updating, or deleting rows in any standard SYSPRO table.
  - Issues could arise due to constraint validation and foreign keys.
  - The most common issue is defining a column as NOT NULL and not defining a default (see the next point).

- NULL and/or Default values
  - When a SYSPRO program inserts a row to a table it will not 'know' about your User Columns and will therefore not supply a value. It is up to you to either define the User Column as allowing NULLs or alternatively supply an appropriate default constraint.
  - We recommend allowing NULLs in User Columns.
  - Your User Column must not prevent standard SYSPRO applications from inserting rows into SYSPRO tables.
- Maximum row size
  - SQL Server allows a table to have a maximum of 8060 bytes per row. There are some exceptions depending on the data types of the columns in the row.
  - Always consider that in future versions of SYSPRO we may add a significant number of columns to any table and therefore you should not approach this limit when adding User Columns.
  - No standard SYSPRO 8 table currently approaches this 8060-byte maximum.
- Collation
  - When adding a User Column for alphanumeric data types (such as char, varchar, varchar(max)), we strongly suggest that your User Column has the same collation as the remainder of the alphanumeric columns in the SYSPRO database.
  - Failure to use the same collation may lead to unexpected results when attempting to access standard SYSPRO columns and your User Columns in the same SQL statement. This includes conditions, JOINS, string concatenation etc.
- User Columns should not be created with an Identity property
  - An Identity property means that a number is automatically generated (typically 1 upwards) when a row is added to the table. You should not add identify columns to SYSPRO tables.
  - One of the reasons for this is that database upgrades will not necessarily retain the generated Identity values and could cause an upgrade to fail.

## USER COMPUTED COLUMNS - ALLOWED

User Computed Columns in SYSPRO tables describes the requirement to add computed columns to a standard SYSPRO table.

A computed column is a virtual column that is not physically stored in the table, unless the column is marked PERSISTED. A computed column expression can use data from other columns to calculate a value for the column to which it belongs.

Generally adding User Computed Columns has no effect on the running of the SYSPRO application.

However, this assumes that you have considered the following:

- The naming convention of any User Computed Columns must not conflict with current or possible future SYSPRO column names.

You could achieve this by reviewing the naming conventions for standard SYSPRO columns. These are described against the previous topic 'User Columns' and will not be repeated here.

There are no standard computed columns in SYSPRO tables.

### GUIDELINES WHEN ADDING USER COMPUTED COLUMNS

Due to the above we recommend that any User Computed Columns should have a different naming convention from the standard SYSPRO columns. This is described against the subject 'User Columns' and will not be repeated here.

In addition, you should ensure that as standard columns are changed using SYSPRO applications, that only valid User Computed Columns can be generated. In other words, the action of the system generating the User Computed Column must not cause any exception or error in the database.

Adding many User Computed Columns could affect the performance of regular SYSPRO applications and should be avoided.

### USER INDEXES - ALLOWED

User Indexes on SYSPRO tables describes the requirement to add additional alternate indexes, over and above standard SYSPRO indexes, to a standard SYSPRO table. When applied appropriately User Indexes are allowed and, unless an excessive number of indexes are added, should only have a small overhead when inserting or changing data on a table.

However, this assumes that you have considered the following:

- The naming convention of any User Indexes must not conflict with current or possible future SYSPRO primary key or index names. You could achieve this by reviewing the naming conventions for standard SYSPRO key and indexes:
  - Standard primary keys have a name that is generated by concatenating the table name and the word 'Key'.
  - Standard alternate indexes have a name that is generated by concatenating the table name plus the phrase 'Idx', followed by a short mnemonic describing the index using a camel case convention.
- Unless constructed appropriately, User indexes have the possible side effect of causing a duplicate when new data is inserted, or existing data is changed.
  - For example, if you added a User Index on the 'Name' column in the Customer table ('ArCustomer') the index may be applied successfully (as perhaps no duplicate names currently exist). However, at some time in the future you may attempt to add a customer with a duplicate name using a SYSPRO application. SQL Server would return a constraint violation on the insert statement. This could cause inadvertent errors to be returned to the user or cause other problems when using SYSPRO.

- All User Indexes must be designed to ensure that duplicates are never generated.
- Custom form tables in SYSPRO 8 that end with a '+' are variable in nature.
  - The column names in custom form tables are described by the end-user. As the custom form designer utility is used columns are added, changed, or even deleted dynamically.
  - If you have added a User Index to a custom form table and you then use the custom form field designer to maintain or delete the custom field referenced by the User Index you may receive unexpected errors or problems.

### GUIDELINES WHEN ADDING USER INDEXES

Due to the above we recommend that any User Indexes should have a different naming convention from the standard SYSPRO indexes.

As all SYSPRO primary key and alternate index names are prefixed with the table name and only consist of uppercase or lowercase characters, a suggestion could be that User Indexes are added using the prefix 'Usr\_' followed by the name of the table followed by the name of the index.

By using a suitable naming convention (such as that described above) you can avoid possible future conflict with SYSPRO indexes or other constraints.

In addition to the above naming convention issue, you should also be aware of the following before adding User Indexes:

- Must not cause duplicates.
  - The columns that make up a User Index must ensure that it is always unique and that inserting or updating rows must never cause duplicates to be generated.
  - The simplest technique that can be used to guarantee uniqueness is one that standard SYSPRO indexes use. Each User Index should contain all columns from the standard SYSPRO primary key (and may optionally contain additional columns as required). As the Primary key is guaranteed to be unique, using this technique will ensure that your User Index is also unique.
  - Failure to follow this guideline will almost certainly lead to application failures – it is your responsibility to ensure this does not occur.
- Keep User Indexes to the minimum number required.
  - There is an overhead when you have added a User Index and a SYSPRO application inserts new rows or changes data on columns referenced by the User Index.
  - If you do add an excessive number of User Indexes (the exact definition of this will vary with the situation) then SYSPRO applications may be slowed significantly when data is inserted or updated. You should test your SYSPRO environment to ensure that the system has not been slowed excessively when adding many User Indexes to a single table.  
Note that this also affects Custom form tables.
  - Therefore, you should only add User Indexes when necessary and avoid an excessive number of User Indexes against a single table.

- As a guideline, adding one or two User Indexes typically has little effect. If you add more than this, you should verify that the overhead does not slow SYSPRO applications significantly.

## USER FOREIGN KEYS - ALLOWED

User Foreign Keys on SYSPRO tables describes the requirement to add additional foreign keys, over and above standard SYSPRO foreign keys, to a standard SYSPRO table.

When applied appropriately User Foreign Keys are allowed and should only have a minimal overhead when working with SYSPRO data.

In addition, you must always use the 'WITH NOCHECK' clause when adding User Foreign Keys as SYSPRO applications handle data integrity using business logic described in Business Objects and not using database constraints such as foreign keys.

There are basically two types of User Foreign Keys that are discussed here:

1. Foreign keys from a User Table into a standard SYSPRO table.
2. Foreign keys from a standard SYSPRO table into a User Table.

In both cases you should consider the following:

- The naming convention of any User Foreign Keys must not conflict with current or possible future SYSPRO foreign key names. You could achieve this by reviewing the naming conventions for standard SYSPRO foreign keys:
  - Standard foreign keys have a name that is generated by concatenating 'Syspro\_FK\_' followed by the source table and then an underscore and then the target table.
  - A suggestion is to prefix User Foreign Keys with 'Usr\_'.
- Foreign keys from a User Table to a standard SYSPRO table or the other way around must not enforce constraints – you must have a 'WITH NOCHECK' clause.
- Although both of the following are relatively rare, you should be aware of the following:
  - In future versions of SYSPRO some columns may be renamed. We try and avoid this unless necessary hence it's relative rare.
  - In future versions of SYSPRO the columns that make up the construct of the primary key may change. Again, we only do this if necessary and hence it's relatively rare.
- Custom form tables end with a '+' and are variable in nature.
  - The column names in custom form tables are described by the end-user.
  - Therefore, you should try and avoid referencing these columns in User Foreign Keys else using the custom form designer may give unexpected messages.

## GUIDELINES WHEN ADDING USER FOREIGN KEYS

Due to the above we recommend that any User Foreign Keys should have a different naming convention from the standard SYSPRO foreign keys.

In addition to the above naming convention issue, you should also be aware of the following before adding User Foreign Keys:

- Must be defined with a 'WITH NOCHECK' clause.
  - Foreign keys from a User Table to a standard SYSPRO table or the other way around must not enforce constraints – you must have a 'WITH NOCHECK' clause.
  - This means 'in effect' that User Foreign Keys into/out of standard SYSPRO tables are used purely to define the relationships between tables and not to enforce constraints on the actual rows in the tables.
- As it's possible for future versions of the SYSPRO database to rename columns and that the primary keys may change (however unlikely) you should be aware that in future your User Foreign Keys may not be applicable.
- Custom form tables that end with a '+' are variable in nature.
  - The column names in custom form tables are described by the end-user.
  - Therefore, you should try and avoid referencing these columns in User Foreign Keys else using the custom form designer may give unexpected messages.

## USER STORED PROCEDURES – ALLOWED

User Stored Procedures in SYSPRO tables describes the requirement to add stored procedures to a standard SYSPRO table.

There are no standard stored procedures in SYSPRO tables. As described earlier in this document, SYSPRO Business Objects encapsulate all our business logic and provide an open and extensible programming interface. Wherever possible we suggest that you use Business Objects for any database access rather than accessing the database directly.

If you require to add User Stored Procedures, they will generally have no effect on the running of the SYSPRO application. However, this assumes that you have considered the following:

- The naming convention of any User Stored Procedures must not conflict with current or possible future SYSPRO database object names.

## GUIDELINES WHEN ADDING USER STORED PROCEDURES

If you require to add a User Stored Procedure to a standard SYSPRO database, you should consider the following:

- To avoid potential problems with future SYSPRO application upgrades, we recommend that any User Stored Procedures should have a different naming convention from any standard SYSPRO database objects.
- In future versions of SYSPRO some columns may be renamed. We try and avoid this unless necessary, hence it's relatively rare. However, User Stored Procedures based on SYSPRO tables may reference one or more renamed columns in future.

## USER TRIGGERS - ALLOWED

User Triggers on SYSPRO tables describes the requirement to add a SQL TRIGGER to a standard SYSPRO table. The standard SYSPRO database does not contain any SQL TRIGGERS.

Before you add a TRIGGER to the SYSPRO database you should consider the condition that will cause the trigger to be invoked together with the actions required and whether they can be performed by the standard SYSPRO application.

SYSPRO has a powerful and configurable set of capabilities for firing triggers of various types – many of which go way beyond the capability of a SQL Server TRIGGER. All built-in capabilities described here are designed to be future proof – allowing for simple software upgrades in future.

These include:

- **eSignatures**
  - The eSignatures feature in SYSPRO combines three main capabilities:
    - **Authentication**  
Preventing a transaction unless authenticated.
    - **Logging**  
Summary and detail logging of each transaction including configuring which fields/variables are to be logged.
    - **Triggering**  
Upon successful completion of a SYSPRO transaction, one or more triggers can be fired.
  - eSignature Triggers can include:
    - **Email**  
Send an email to one or more recipients. System-wide and Transaction specific variables are available.
    - **Run a VBScript**  
Allows a VBScript to be invoked. This can perform virtually any task including invoking Business Objects.
    - **Run any program**  
Allows developer code to be invoked with parameter passing.
    - **Write to message inbox**  
Allows you to write information to a special message inbox.
    - **Run an SRS report**  
Allows you to invoke any SYSPRO report. The report can physically print a document, send the report by email, export in a variety of formats (such as PDF or Excel) and can be archived.
  - eSignatures can be configured system-wide, by company, by role, by group or by operator.
  - Multiple triggers can be fired for each environment, condition, and transaction.
  - eSignatures can be configured to be active on many user defined conditions.
  - There are over 800 eSignature transactions that can be configured in SYSPRO 8.



## ▪ **SYSPRO Triggers**

- Various SYSPRO programs have 'trigger points' where an application can be invoked, or other actions performed.

These include:

- **Run a standard SYSPRO program**  
This can cause another SYSPRO program to be invoked allowing the operator to interact with a relevant program.
- **Run any program**  
Allows developer code to be invoked with parameter passing.
- **Run a customized report**  
Allows a Report Writer report to be invoked passing parameters about the cause of the trigger. These reports can produce physical reports, generate export files of various types and even update data in one or more tables.
- **Write to message inbox**  
Allows you to write information to a special message inbox.
- There are about 200 applications that have trigger points in SYSPRO 8.

## ▪ **SYSPRO Events**

- Various events can be configured to fire triggers or actions.  
An event is something like 'Bank balance below a parameter'. Whenever the event occurs (regardless of the actual program making the change) one or more triggers can be fired.

These include:

- **Email**  
Send an email to one or more recipients. System-wide and Transaction specific variables are available.
- **Run a standard SYSPRO program**  
This can cause another SYSPRO program to be invoked allowing the operator to interact with a relevant program.
- **Run any program**  
Allows developer code to be invoked with parameter passing.
- **Write to message inbox**  
Allows you to write information to a special message inbox.
- There are over 50 different events that can be configured in SYSPRO 8.

## GUIDELINES WHEN ADDING USER TRIGGERS

If you require to add a User Trigger to a standard SYSPRO table, you should consider the following:

- To avoid potential problems with future SYSPRO application upgrades, we recommend that any User Triggers should have a different naming convention from any standard SYSPRO database objects.

- In future versions of SYSPRO some columns may be renamed. We try and avoid this unless necessary, hence it's relatively rare. However, User Triggers based on SYSPRO tables may reference one or more renamed columns in future.
- User Triggers should be carefully developed to prevent exceptions being raised which could cause the SYSPRO transaction firing the SQL TRIGGER to be rolled back. This could cause confusion to end-users and may give unexpected messages.
  - Consider outputting a status or log to a separate table indicating the success or otherwise of a User Trigger – this can then be used to determine if there have been any problems.
- User Triggers should be carefully developed to avoid adding excessive overhead to regular SYSPRO transactions.
  - As a User Trigger can perform virtually any database access, it's possible that a poorly developer SQL TRIGGER can cause excessive performance degradation to the SYSPRO application.
- User Triggers that in any way access data from standard SYSPRO tables may cause deadlocks or other contention issues.
  - In many cases, a User Trigger reads data from a standard SYSPRO table and, based on the values returned, performs some other function. You should be careful when reading data from a User Trigger as the rows may be locked – either due to the SYSPRO transaction causing the trigger or another SYSPRO transaction initiated by a different user. When developing User Triggers, you should consider adding a 'WITH (NOLOCK)' clause to your SELECT statements causing a dirty read and preventing possible deadlocks.
  - **Deadlocks caused by User Triggers have been found to be one of the most common problems caused by User Customization in SYSPRO databases.**
- Custom form tables that end with a '+' are variable in nature.
  - The column names in custom form tables are described by the end-user.
  - Therefore, you should try and avoid referencing these columns in User Triggers else using the custom form designer may give unexpected messages, or if someone uses the custom form designer your User Trigger may fail unexpectedly.

## USER VIEWS - ALLOWED

User Views on SYSPRO tables describes the requirement to add a View to a standard SYSPRO base table. The standard SYSPRO database does not contain any views.

A User View creates a virtual table whose contents (columns and rows) are defined by a query. For example, a view can be used for the following purposes:

- To focus, simplify, and customize the perception each user has of the database.
- As a security mechanism by allowing users to access data through the view, without granting the users permissions to directly access the underlying base tables.

- To provide a backward compatible interface to emulate a table whose schema/structure has changed.

When applied appropriately, User Views are allowed and should only have a minimal overhead when working with SYSPRO data. SYSPRO programs will not use or reference the User View. However, the Report Writer can be configured to report on a User View.

In some cases, a User View may be used by the SQL query optimizer even when not explicitly referenced so may even improve the performance of standard SYSPRO applications.

You should consider the following:

- The naming convention of any User View must not conflict with any other SYSPRO object whether now or into the future. You can achieve this by reviewing the current naming conventions and using a different naming convention for your User Views.
- In future versions of SYSPRO some columns may be renamed. We try and avoid this unless necessary hence it's relatively rare. However, User Views based on SYSPRO tables may reference one or more renamed columns in future.
- Custom form tables that end with a '+' are variable in nature.
  - The column names in custom form tables are described by the end-user.
  - Therefore, you should try and avoid referencing these columns in User Views else using the custom form designer may give unexpected messages.

## GUIDELINES WHEN ADDING USER VIEWS

Due to the above we recommend that any User Views should have a different naming convention from any standard SYSPRO objects.

In addition to the above naming convention issue, you should also be aware of the following before adding User Views:

- SYSPRO applications do not reference User Views and therefore should only be created for third party database access to standard SYSPRO tables.
- In future versions of SYSPRO some columns may be renamed. We try and avoid this unless necessary, hence it's relatively rare. However, User Views based on SYSPRO tables may reference one or more renamed columns in future.
- When creating User Views, you should reference columns names explicitly rather than use an '\*' to match all columns. This ensures that if the base table changes in future that your User View will remain unchanged.
- Custom form tables that end with a '+' are variable in nature.
  - The column names in custom form tables are described by the end-user.
  - Therefore, you should try and avoid referencing these columns in User Views else using the custom form designer may give unexpected messages.

## USER CHECK CONSTRAINTS – ALLOWED BUT NOT RECOMMENDED

User Check Constraints on SYSPRO tables describes the requirement to add check constraints to a standard SYSPRO table.

The standard SYSPRO database does not contain any check constraints.

You should always consider using standard SYSPRO business logic encapsulated in Business Objects before making any changes to the SYSPRO database. The SYSPRO application uses Business Objects to provide business logic rather than using check constraints in the database.

A User Check Constraint on a standard SYSPRO column could be used to limit the possible values in a column. You may want to consider using a User Trigger instead of a User Check Constraint as this provides more control in the event of a check constraint violation.

When applied appropriately, User Check Constraints are allowed (but not recommended) and should only have a minimal overhead when working with SYSPRO data.

### GUIDELINES WHEN ADDING USER CHECK CONSTRAINTS

If you require to add a User Check Constraint to a standard SYSPRO table, you should consider the following:

- To avoid potential problems with future SYSPRO application upgrades, we recommend that any User Check Constraints should have a different naming convention from any standard SYSPRO database objects.
- In future versions of SYSPRO some columns may be renamed. We try and avoid this unless necessary, hence it's relatively rare. However, User Check Constraints based on SYSPRO tables may reference one or more renamed columns in future.
- User Check Constraints (by definition) may cause standard SYSPRO applications to fail when inserting or changing data.
  - This may not be immediately obvious to the operator and it is completely your responsibility to ensure that any exceptions raised are understood by the operators using the software.
  - For this reason, it is recommended that you rather add a User Trigger and detect the constraint condition. Then, where relevant, you could log any exceptions (and reject the database insert/update if required).
  - Alternatively, by appropriate use of a VBScript, the SYSPRO application could be changed to apply the same logic as the User Check Constraint. This is preferable to changes to the SYSPRO database.
- Custom form tables that end with a '+' are variable in nature.
  - The column names in custom form tables are described by the end-user.
  - Therefore, you should try and avoid referencing these columns in User Check Constraints else using the custom form designer may give unexpected messages.

## USER SCHEMAS – ALLOWED

A User Schema on SYSPRO tables describes the requirement to add additional schemas, over-and-above the 'dbo' schema used by SYSPRO, on one of the SYSPRO databases.

The topic about Database Schema earlier in this document - describes that a schema is a container for objects within the database. SYSPRO objects reside in the schema named 'dbo'.

You can create User Schemas and add your own objects into your own schema. However, you should not 'move' standard SYSPRO objects into your schema as all standard SYSPRO objects must reside in a single schema named 'dbo'.

## GUIDELINES WHEN ADDING USER SCHEMAS

If you require to add a User Schema to a standard SYSPRO table, you should consider the following:

- User Schema – Object naming conventions
  - Although you can create a User Schema and within that insert objects that have the same name as standard SYSPRO objects, you should do this with caution.
  - The majority of SYSPRO applications will ignore the User Schema – and therefore any user objects within the User Schema. However, it is possible that one or more SYSPRO applications (such as utility programs) may not take account of the Schema Name when viewing or testing for the existence of objects with the same name as standard SYSPRO objects.
  - Another reason to avoid creating objects in your User Schema with the same name as standard SYSPRO objects is that this is often very confusing to administrators, developers and support personnel. Only use objects with the same name when you really intend to create this potential confusion.
- All SQL Logins used when logging into SYSPRO must have a default schema of 'dbo'. This is to ensure that the User Schema is not accidentally used in SYSPRO applications.

# Configuring SYSPRO to work with SQL Server

This topic describes various options when setting up and configuring SYSPRO to work with SQL Server.

Most of these options are defined in the **Setup Options > System Setup > SQL Tab**.

See the example below:

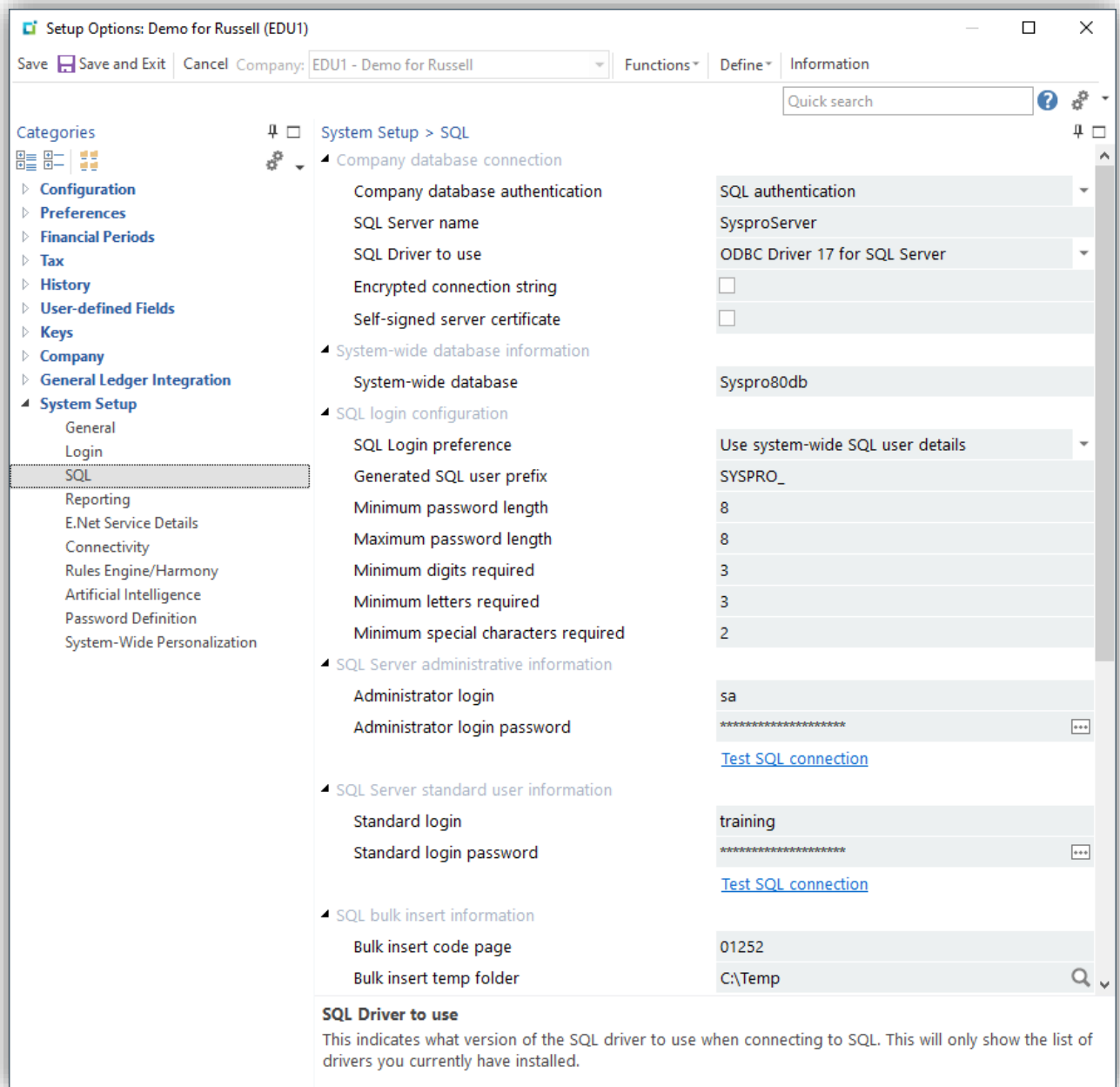


Figure 5: Setup Options > System Setup > SQL Form

## COMPANY DATABASE AUTHENTICATION

There are two main methods that applications can use to authenticate connections to SQL Server. SYSPRO can be configured to use either method.

The following information is taken from the Microsoft SQL Server documentation located at:

<http://msdn.microsoft.com/en-us/library/ms144284.aspx>

*The text shown with an orange font below is taken from the above-mentioned web page.*

*During setup of SQL Server, you must select an authentication mode for the Database Engine. There are two possible modes: Windows Authentication mode and Mixed mode. Windows Authentication mode enables Windows Authentication and disables SQL Server Authentication. Mixed mode enables both Windows Authentication and SQL Server Authentication. Windows Authentication is always available and cannot be disabled.*

## WINDOWS AUTHENTICATION

*When a user connects through a Windows user account, SQL Server validates the account name and password using the Windows principal token in the operating system. This means that the user identity is confirmed by Windows. SQL Server does not ask for the password, and does not perform the identity validation.*

*Windows Authentication is the default authentication mode, and is much more secure than SQL Server Authentication. Windows Authentication uses Kerberos security protocol, provides password policy enforcement with regard to complexity validation for strong passwords, provides support for account lockout, and supports password expiration.*

*A connection made using Windows Authentication is sometimes called a trusted connection, because SQL Server trusts the credentials provided by Windows.*

When you wish to use 'Windows authentication' to connect to SQL Server from SYSPRO, configure your **Setup Options** program (*Setup Options > System Setup > SQL*) to have the database connection 'Authentication' set to 'Windows authentication'.

## SQL SERVER AUTHENTICATION

*When using SQL Server Authentication, logins are created in SQL Server that are not based on Windows user accounts. Both the user name and the password are created by using SQL Server and stored in SQL Server.*

*Users connecting using SQL Server Authentication must provide their credentials (login and password) every time that they connect.*

*When using SQL Server Authentication, you must set strong passwords for all SQL Server accounts.*

When you wish to use 'SQL Server authentication' to connect to SQL Server from SYSPRO configure your **Setup Options** program (*Setup Options > System Setup > SQL*) to have the database connection 'Authentication' set to 'SQL authentication'.

## HOW DOES SYSPRO CONNECT TO SQL SERVER?

The following topic summarizes the process by which SYSPRO connects to SQL Server and how it determines the correct login method.

The fields in the **Setup Options > System Setup > SQL** form – will be described more fully after this overview.

When you run SYSPRO, the initial application reads the `IMPACT.INI` file defined in your `WORK` folder. See the following fragment from a sample `IMPACT.INI`:

```
[Database Settings]
SQLLGN=SQLSERVER
SQLSSN=SysproServer|
SQLDBN=Syspro80db
SQLADM=1gM7kth7LhzpsPzJpTBo9bviJ1tad4jZo+GLNUztMmc9
SQLSTD=q4wq0dj1I00qw0Am8x56CvNo9eBijjvxxk6TkP6r/OTPI
SQLBLK=C:\Temp
SQLCPG=01252
SQLCSO=2,0,0,0
```

The `SQLLGN=` entry contains the 'Authentication' setting as defined in the **Setup Options** program.

The valid values are:

- SQLSERVER (uses SQL authentication)
- WINDOWS (uses Windows authentication)

If Windows authentication is selected, then the Windows user associated with the SYSPRO Communication Service will be used for all SQL Server access. This user must have SQL Server administrative privileges to the SYSPRO system-wide and company databases if the SYSPRO applications needs to maintain the database – for example when using the custom form designer or when applying a minor database update.

If SQL authentication is selected, then SYSPRO will proceed to login to SQL Server using the SQL Server standard user information as defined in the **Setup Options** program. The standard SQL login will be used by the majority of SYSPRO applications and requires **db\_datareader** and **db\_datawriter** access to both the system-wide and company specific databases.

If an administrative task is required (such as using the custom form designer where a new column may need to be created) the SQL Server administrative information will be used to elevate the privileges whilst the database maintenance is being performed. As soon as the task has completed the administrative SQL login will be disconnected – reverting to the standard SQL login as before.

If required, each SYSPRO operator can have their own SQL login, instead of using the SQL Server standard user information defined system wide. This operator specific SQL login requires **db\_datareader** and **db\_datawriter** access to both the system-wide and company specific databases - the same as the standard SQL login.

When you exit SYSPRO all SQL connections will be disconnected.



When an operator is viewing the Login Dialog in SYSPRO, the current database will be the system-wide database. In other words, the Windows authenticated user or SQL Server standard login (depending on the authentication method) will have the context of the system-wide database described in the **Setup Options** program (*Setup Options > System Setup > SQL*).

When the operator attempts to log into SYSPRO the **'SysproAdmin'** table in the system-wide database is used to determine the company specific database name. The context will be changed to the relevant company specific database and all SQL statements issued until you exit the company (or SYSPRO) will be relative to the company database. If the operator uses the main menu option to return to the Login Dialog to allow another company to be entered the context is changed back to the system-wide database.

## SETUP OPTIONS > SYSTEM SETUP > SQL FORM REFERENCE

The following topic describes the SQL form options of the **Setup Options** program and available values. See the sample screen shot:

Section	Field Name	Value
Company database connection	Company database authentication	SQL authentication
	SQL Server name	SysproServer
	SQL Driver to use	ODBC Driver 17 for SQL Server
	Encrypted connection string	<input type="checkbox"/>
	Self-signed server certificate	<input type="checkbox"/>
System-wide database information	System-wide database	Syspro80db
SQL login configuration	SQL Login preference	Use system-wide SQL user details
	Generated SQL user prefix	SYSPRO_
	Minimum password length	8
	Maximum password length	8
	Minimum digits required	3
	Minimum letters required	3
SQL Server administrative information	Administrator login	sa
	Administrator login password	*****
		<a href="#">Test SQL connection</a>
SQL Server standard user information	Standard login	training
	Standard login password	*****
		<a href="#">Test SQL connection</a>
SQL bulk insert information	Bulk insert code page	01252
	Bulk insert temp folder	C:\Temp

## SETUP OPTIONS > SYSTEM SETUP > SQL: COMPANY DATABASE AUTHENTICATION

As described previously the 'Company database authentication' option allows you to configure how you wish SYSPRO to connect to SQL Server.

If the SQL Server instance has been configured to use only 'Windows authentication' then you must select 'Windows authentication' in the **Setup Options** program. Even if your SQL Server instance allows SQL Server authentication, you can also select Windows authentication in SYSPRO. This means that all SQL Server access from within the SYSPRO application will use the Windows user defined against the SYSPRO Communication Service. This login must be assigned SQL Server administrative privileges.

If the SQL Server instance has been configured to allow 'Windows authentication and SQL Server authentication' then you can select SQL authentication in the **Setup Options**. This means that the SQL Server login and password defined in the **Setup Options** will be used to connect to SQL. SYSPRO requires a standard SQL Server user with **db\_datareader** and **db\_datawriter** access to both the system-wide and company specific databases. In addition, SYSPRO requires an administrative SQL Server user that will be used to perform 'administrative tasks'.

## SETUP OPTIONS > SYSTEM SETUP > SQL: SQL SERVER NAME

You must define the name of the SQL Server instance to which you wish to connect.

## SETUP OPTIONS > SYSTEM SETUP > SQL: SQL DRIVER TO USE

You must define the ODBC Driver to be used to communicate between SYSPRO and SQL Server. SYSPRO communicates with SQL Server using ODBC drivers provided by Microsoft. These provide standardized, robust, and high-performance interfaces to SQL Server.

SYSPRO 8 2020 R1 supports three different ODBC Drivers:

- SQL Server
- ODBC Driver 13 for SQL Server
- ODBC Driver 17 for SQL Server

The first driver (simply named **SQL Server**) ships as part of Windows and is known as a Windows Data Access Component (WDAC) – it has provided ODBC access to SQL Server for applications such as SYSPRO for many years. However, Microsoft have recently indicated that new software should no longer use this driver, partly because some features (e.g. relating to encryption) are not fully available with the **SQL Server** (WDAC) driver.

For this reason, SYSPRO 8 2020 R1 has been enhanced to allow more recent ODBC drivers to be specified in the **Setup Options**. It is recommended that, if available, you use the latest ODBC Driver, such as **ODBC Driver 17 for SQL Server**. The **Setup Options** only shows the drivers installed on your SYSPRO application server.

**Note:** You may have to download and install the required ODBC driver if it is not currently installed on your SYSPRO application server.

## SETUP OPTIONS > SYSTEM SETUP > SQL: ENCRYPTED CONNECTION STRING

If required, you can check this item to ensure that SYSPRO uses an encrypted connection string. For more information see the encryption topic: [Data Encryption in Motion](#).

## SETUP OPTIONS > SYSTEM SETUP > SQL: SELF-SIGNED SERVER CERTIFICATE

If required, you can check this item to configure SYSPRO to use a self-signed sever certificate. For more information see the encryption topic: [Data Encryption in Motion](#).

## SETUP OPTIONS > SYSTEM SETUP > SQL: SYSTEM-WIDE DATABASE

You must define the name of the system-wide database. This is the database that contains system-wide tables including the '**SysproAdmin**' table that contains the cross-reference between the company id and the database name and its collation.

## SETUP OPTIONS > SYSTEM SETUP > SQL: SQL LOGIN CONFIGURATION

You must configure how you want the regular SYSPRO database access to be managed. The choices are:

- Use system-wide SQL user details
- Use operator specific SQL user details
- Use generated SQL user details

If you select the first option (**Use system-wide SQL user details**), this means that the standard SQL username and password will be used for all users logging into SYSPRO. See the option 'SQL Server standard user information' below.

If you use the second option (**Use operator specific SQL user details**) you can specify against each operator code a specific SQL username and password. When the user logs into SYSPRO, the specific SQL username and password for that operator will be used. If there is not a specific SQL username and password configured for the operator, the system will revert to using the SQL Server standard user information.

For more information about setting up this environment see the topic below headed: [Defining a SQL Server login against each operator \(optional\)](#).

If you use the third option (**Use generated SQL user details**) then the remainder of the fields immediately below the dropdown will be used to generate a SQL username and a strong password:

- Generated SQL user prefix
- Minimum password length
- Maximum password length
- Minimum digits required
- Minimum letters required
- Minimum special characters required

## SETUP OPTIONS > SYSTEM SETUP > SQL: SQL SERVER ADMINISTRATIVE INFORMATION

You must define the SQL Server login and password of a SQL Server account that has administrative privileges to both the system-wide and company specific databases.

This administrative account requires the following privileges:

- **DATABASE CREATE**  
Ability to add a new database, create and drop tables, columns, indexes, foreign keys etc.
- **BULK INSERT**  
Ability to execute a BULK INSERT statement to upload data to the database.

Examples where the administrative login is used:

- When using the Custom Forms Designer and a new Table or Column is to be added to the SYSPRO database, or an existing column is to be deleted or renamed. All Custom Form tables affected by this are suffixed with '+'.  
▪ When installing a new release of SYSPRO and performing a minor database update - this SQL login will be used to add additional Tables, Indexes, Columns, or other objects as required.
- When using a SQL health dashboard or diagnostic program, as these can require elevated permissions to query system and cross-database parameters.

You must set strong passwords for all SQL Server accounts. You will often use a SQL Server account that has a fixed server 'sysadmin' role – such as 'sa' - but other accounts are acceptable if they have the correct privileges.

## SETUP OPTIONS > SYSTEM SETUP > SQL: SQL SERVER STANDARD USER INFORMATION

You must define the SQL Server login and password of a SQL Server account that has data read and write privileges to both the system-wide and company specific databases.

This standard SQL account requires the following privileges:

- **db\_datareader**  
Ability to read all rows from any table in the database.
- **db\_datawriter**  
Ability to insert or change any data in any row from any table in the database.

You must set strong passwords for all SQL Server accounts. You may not use the same SQL Server account for both standard and administrative users.

## SETUP OPTIONS > SYSTEM SETUP > SQL: SQL BULK INSERT INFORMATION

You must define a bulk insert code page and temporary folder that will be used when SYSPRO applications require to perform BULK INSERT operations.

For example:

- Refreshing the data dictionary when new software is installed uses the bulk insert SQL function to perform the update as quickly as possible.
- Running an MRP Requirement Calculation can require a bulk insert of large volumes of data into snapshot and suggested tables.

If you are using English or Western European languages, then leave the bulk insert code page as 01252. However, for other database collations you can select a relevant code page to be used during the BULK INSERT process.

The bulk insert process formats data into a plain text file with a Tab Separated format and then issues a BULK INSERT statement in SQL Server – in effect importing the data from the file system. This is very quick, hundreds or thousands of times time faster than inserting data row-by-row.

The bulk insert temp folder must be a folder that is accessible from both the SYSPRO application server and the server on which SQL Server is running. Often you will require entry of a UNC path to a temporary folder for this purpose – especially in a 3-tier environment.

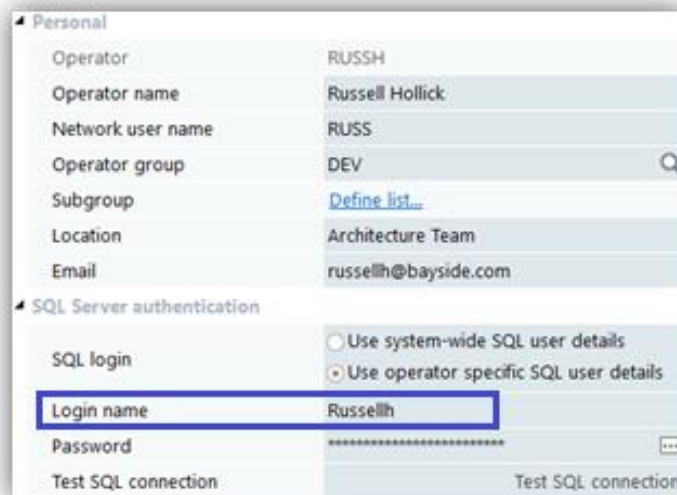
## DEFINING A SQL SERVER LOGIN AGAINST EACH OPERATOR (*OPTIONAL*)

As mentioned previously, if you are using SQL Server authentication, you must define a standard and administrative SQL login in the **Setup Options** program (*Setup Options > System Setup > SQL*).

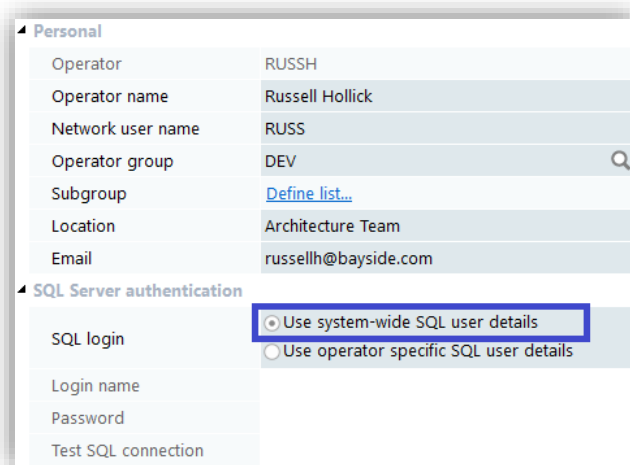
In addition, as an operator logs into SYSPRO, you can override the standard SQL login and use an operator specific SQL Server login and password – this is defined against the operator code.

The following lists the advantages and disadvantages of having separate SQL Server login and password per operator:

- **Defining a SQL Server login and password against each operator**
  - This assumes that you have assigned a separate SQL Server login and password against each operator using the **Operator Maintenance** program (i.e. IMPPMP).
  - See the following example where the SQL login name 'Russellh' will be used to connect to SQL Server when this operator logs into SYSPRO:



- The advantage of this method is that each SYSPRO operator can use a different SQL Server login and password and therefore can be given different parameters or defaults in SQL Server. In addition, when tracing SQL Server access, each operator's database access can be isolated – this can help when performing system diagnostics.
- The downside is that if you have many SYSPRO operators then maintenance of all these SQL login and passwords (both in SQL Server and SYSPRO) can be a drain on administrative resources and can lead to less-than-ideal security unless strong passwords are always maintained.
- **Not defining a SQL login and password per operator**
  - The default is that all operators 'use system-wide SQL user details' – as shown below:



- Selecting this option means that the **Setup Options** program will be used to determine how to connect to SQL Server.

# SQL CONNECTION STRINGS

SYSPRO uses ODBC to connect to SQL Server.

**Important:** The following quote is from a Microsoft SQL Server resource (<http://blogs.msdn.com/b/sqlnativeclient/archive/2013/01/23/introducing-the-new-microsoft-odbc-drivers-for-sql-server.aspx>):

*"Microsoft is adopting ODBC as the de-facto standard for native access to SQL Server and Windows Azure SQL Database."*

When SYSPRO connects to SQL Server it creates a connection string. This string requires the SQL Server name and the authentication method – this includes the SQL Server login and password if using SQL authentication.

A sample generated connection string is shown below (with the password hidden):

```
DRIVER={SQL Server}; UID=SysproLogin; PWD=*****;  
Trusted_Connection=no; Server=SysproServer;
```

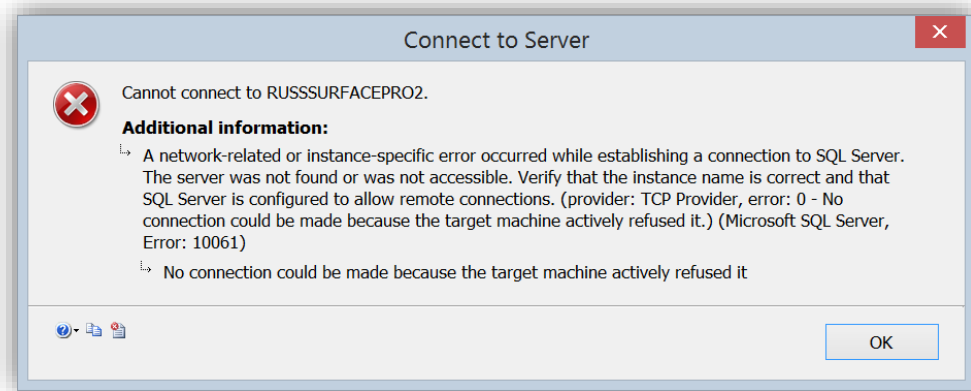
See the topic about ODBC Drivers for more information: [Setup Options > System Setup > SQL: SQL Driver to use](#)

## NO CONNECTION COULD BE MADE BECAUSE THE TARGET MACHINE ACTIVELY REFUSED IT

When trying to login to SYSPRO you may receive a message from SQL Server indicating that it cannot connect to the instance of SQL server with a message saying:

**“No connection could be made because the target machine actively refused it”**

See an example below:

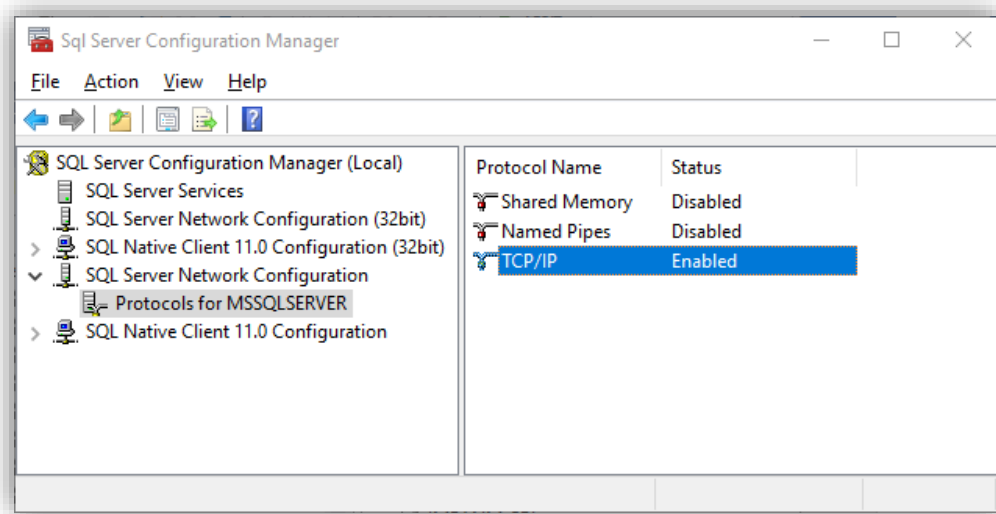


There are many possibilities that may cause this message. However, one of the common ones is where you are trying to use TCP/IP but it has been disabled as a valid connection protocol.

To verify whether this is the cause:

1. Load the SQL Server Configuration Manager
2. Open the 'SQL Server Network Configuration' node
3. Select Protocols for MSSQLSERVER.

See the following image:



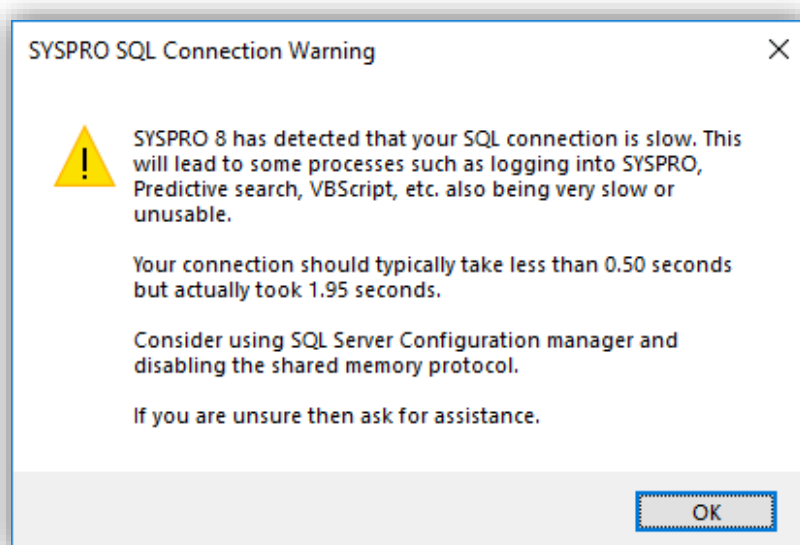
4. Ensure the TCP/IP Protocol Name is 'Enabled'.  
The default on most installations is to disable this Protocol.
5. Once enabled, restart the MSSQLSERVER service – you can do this by clicking on 'SQL Server Services' and right-clicking on 'SQL Server (MSSQLSERVER)' and clicking 'Restart'.

Once enabled and the service restarted you should be able to login to SYSPRO without the message appearing.

## CONNECTION TO SQL SERVER IS SLOW

When SYSPRO connects to SQL Server it detects the time that the connection took to connect. In a correctly configured and working system a connection to SQL Server should be much less than a 1/10<sup>th</sup> of a second (0.1 seconds or less).

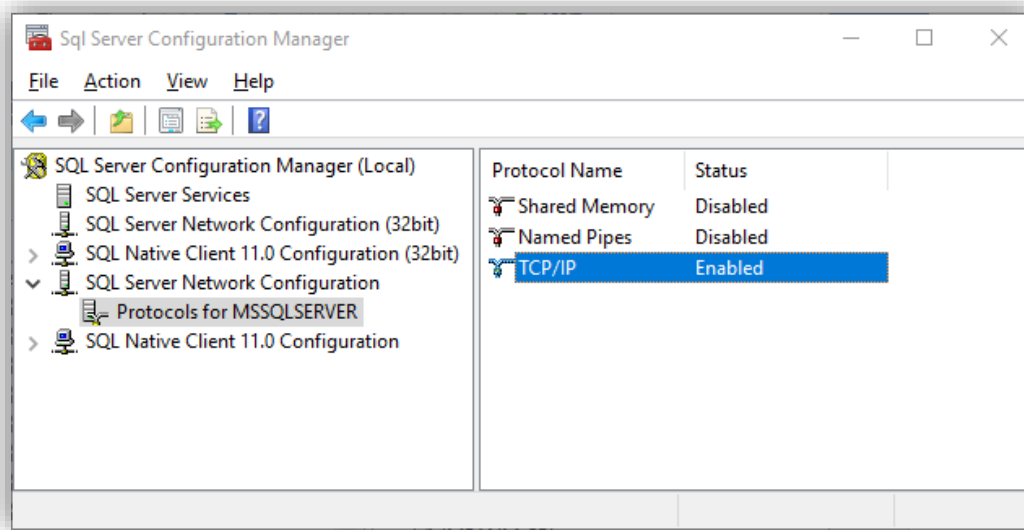
If the login process detects that the connection is unusually slow (takes over 0.5 second) then you will receive a warning like the following:





As the message indicates, this is often an indication that the SQL Server Configuration manager has not been setup appropriately.

Ensure that the Shared Memory and Named Pipes protocols are **Disabled** as shown:



## CREATING A SYSPRO DATABASE USING MANAGEMENT STUDIO

When adding a new SYSPRO company, a wizard is invoked to create the company-specific database, create all the standard SYSPRO tables with appropriate columns and properties, each table's primary keys and alternate indexes together with foreign keys between all related tables.

The wizard then inserts an entry into the system-wide company/database/collation cross-reference table named '**SysproAdmin**'.

Early on during this process the wizard allows you to choose whether you want to create the database yourself or have the wizard create the database for you.

**In SYSPRO 8 we recommend that you create the database yourself.**

There are several important properties against a database that the wizard does not currently allow you to configure. The ability to configure these properties may change in a later version of SYSPRO.

When adding a new database using SQL Server Management Studio there are many settings and options. The following sub-topics will discuss these.

**Important:** From SYSPRO 2020 R2 onwards, the company wizard has been changed so that it does not create the company database. It is important that when you create the company database, the initial size and growth factors are suitable and the collation has been correctly defined and other items relating to data integrity (such as the backup and recovery model) are decided to ensure that the database has been created by a suitable administrator.

## NEW DATABASE – GENERAL TAB

The General Tab is where you define the database name, initial data and transaction log file size and future growth together with their physical file locations. See below:

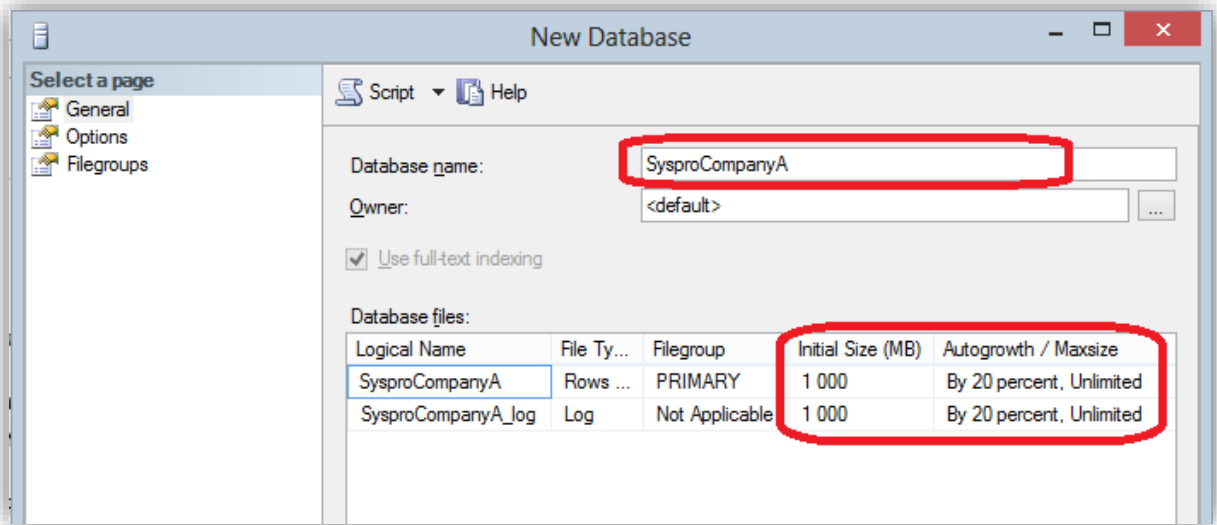


Figure 6: Adding a Database - General Tab

SYSPRO database names must be 18 characters or less. You could consider a database name that describes the company or use the SYSPRO company id (a 1-4-character field consisting of A-Z or 0-9) as part of the database name.

For optimal performance it is important to change the 'Initial Size' and 'Autogrowth/Maxsize' of the data and transaction logs to appropriate values. The default values are typically an initial size of 10 MB with 10% growth. However, a real-world database will be considerably larger than 10 MB – usually at least 1000 MB (approximately 1 GB) and often much larger.

In the scenario where a database has an initial size of 10 MB and growth of 10% defined, the following will occur. As SYSPRO inserts data into the database and the initial size of 10 MB is reached, SQL Server will autogrow the data file by 10%. This means that the data file will now be 11 MB. Quite quickly this will autogrow again by another 10% to 12.1 MB etc. There is an overhead when the autogrow feature enlarges the data file and each autogrow may cause disk fragmentation.

You can see that if the database started at 10 MB and was to grow to 1 GB, increasing by 10% at a time, it would have to autogrow hundreds or even thousands of times – this leads to excessive database resource usage, disk fragmentation and memory usage during the reorganization – all leading ultimately to reduced performance. This occurs not only during the database growth period but the system will continue to perform poorly due to disk fragmentation.

Consider changing the initial database size to something considerably larger than 10 MB. Unless you are creating a small development, test or training database, consider making the initial size 1000 MB (approximately 1 GB) or much more. You may want to change the autogrow percentage to a value larger than 10% - such as 50% or larger, or use a number of megabytes as the growth factor.

## DATA AND TRANSACTION LOG LOCATIONS

Before you move from the General Tab, you should configure the path where the data and transaction log files reside.

See the topic: <http://technet.microsoft.com/en-us/library/ms189563.aspx>.

This includes the following paragraph:

*By default, the data and transaction logs are put on the same drive and path. This is done to handle single-disk systems. However, this may not be optimal for production environments. We recommend that you put data and log files on separate disks.*

Note that you will need to use the slider to scroll to the right (from the database initial size and autogrow options). It's easy to not see the location options as they are initially out of sight.

Reasons for placing the data and transaction log files on different drives include:

- It allows optimal use of RAID configuration and other hardware optimizations.
- It provides resilience in the event of a disk crash or other disk corruption as an appropriate database backup model and backup regime will help you to restore the database quickly with minimal or even no loss of data.
- SYSPRO applications access the database (and therefore the associated data file) with a varied combination of random read, write and update operations, whereas the transaction log file is written sequentially by SQL Server. By locating the data and transaction log files on different physical disks you can gain maximum performance due to the differing types of file access and how they affect physical disk access and disk fragmentation.

The following shows a database being added – note the different disk drive (Path) of the data and transaction log files:

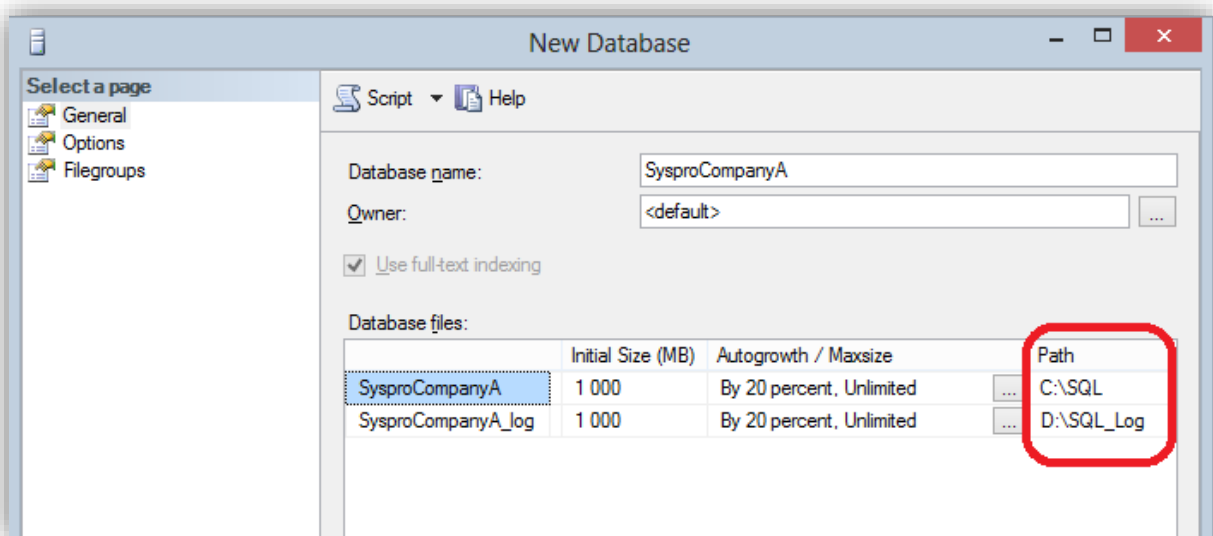


Figure 7: Adding a Database - General Tab - Path

## NEW DATABASE – OPTIONS TAB

The Options Tab is where you define the Collation, Recovery model, Compatibility level and many other properties of the database:

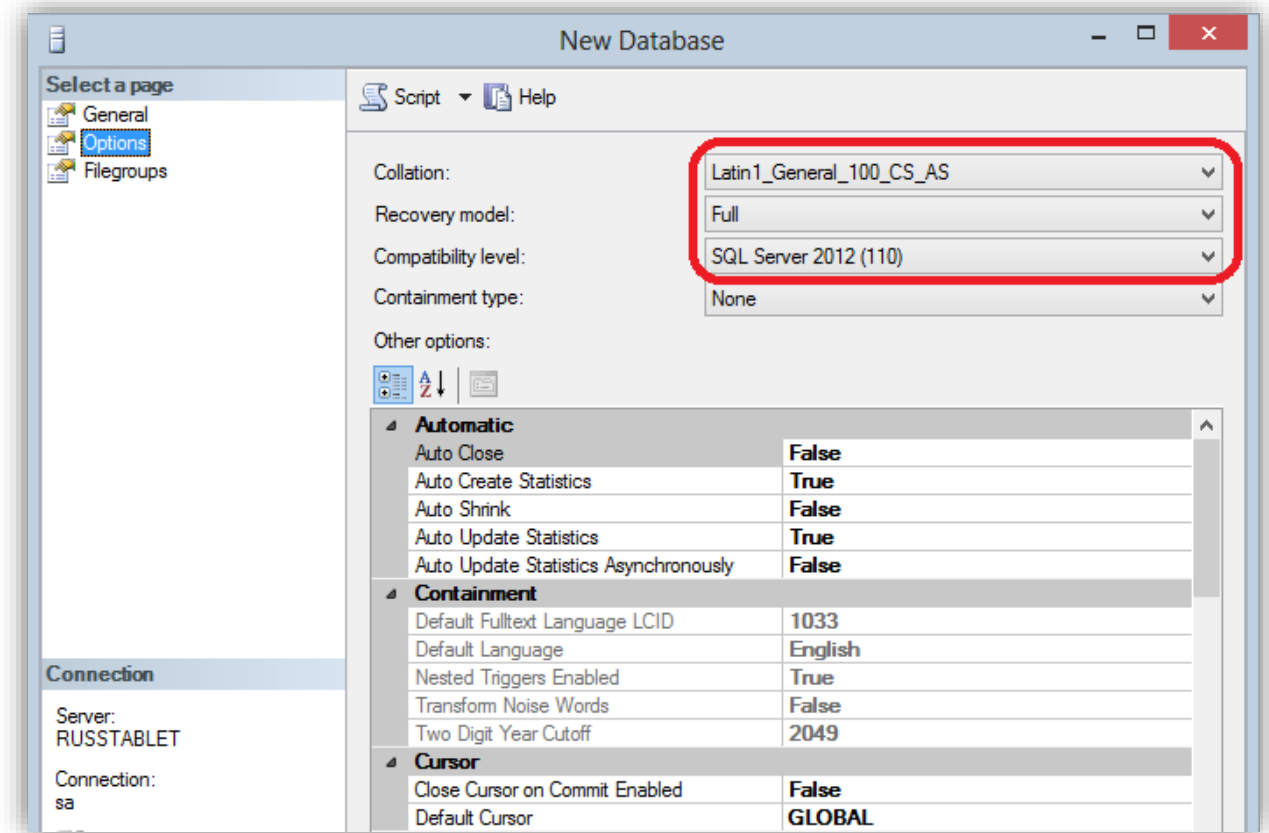


Figure 8: Adding a Database - Options Tab

### COLLATION

It is critical that you select a Case Sensitive and Accent Sensitive collation (or a Binary collation) when adding a SYSPRO database.

This is described more fully earlier in this document under the topic [Collation and Case Sensitivity](#). Remember the suitable collations will either have 'CS\_AS' (or 'BIN' or 'BIN2') as part of their name.

### RECOVERY MODEL

Select an appropriate model after considering how you would like to backup and restore the database in the event of a failure. SYSPRO sites typically use the 'Full' recovery model but this is not mandatory.

See the topic: <http://msdn.microsoft.com/en-us/library/ms189275.aspx>. This includes the following paragraph:

*SQL Server backup and restore operations occur within the context of the recovery model of the database. Recovery models are designed to control transaction log maintenance.*

*A recovery model is a database property that controls how transactions are logged, whether the transaction log requires (and allows) backing up, and what kinds of restore operations are available. Three recovery models exist: simple, full, and bulk-logged. Typically, a database uses the full recovery model or simple recovery model. A database can be switched to another recovery model at any time.*

## COMPATIBILITY LEVEL

When adding a new database use a compatibility level that relates to the current version of SQL Server software. See the [SQL Server Compatibility Levels](#) topic for more information.

## OTHER OPTIONS

The remainder of the database properties are typically left with their default values. If you decide to change any value from their defaults, then carefully consider if this will have any effect when running SYSPRO on this database. Some settings will prevent SYSPRO applications from performing optimally or at all. The remainder of these 'Other options' will not be covered in this document.

## NEW DATABASE – THE MODEL DATABASE

See the topic: <http://technet.microsoft.com/en-us/library/ms186388.aspx>. This includes the following paragraph:

*The model database is used as the template for all databases created on an instance of SQL Server. Because tempdb is created every time SQL Server is started, the model database must always exist on a SQL Server system. The entire contents of the model database, including database options, are copied to the new database. Some of the settings of model are also used for creating a new tempdb during start up, so the model database must always exist on a SQL Server system.*

Due to the above explanation, you might consider making changes to the 'model' database before you add several SYSPRO databases so that they 'inherit' the required defaults.

## WHAT TO DO WHEN SYSPRO CANNOT CONNECT TO SQL SERVER

There may be situations where you are a support person, system administrator or developer and have been given a SYSPRO system to access, however you seem unable to connect to SQL Server.

As an example, perhaps the standard SQL login user and/or password is invalid for some reason.

As discussed earlier, when SYSPRO is loading it reads your `IMPACT.INI` file from the `WORK` folder and uses the entries that start 'SQL' to connect to SQL Server. This includes two entries defining the standard and administrative SQL Server username and password – these entries are both encrypted for obvious security reasons and therefore you cannot manually edit them to enter valid entries.

See the following fragment from an `IMPACT.INI` file:

```
[Database Settings]
SQLLGN=SQLSERVER
SQLSSN=SysproServer
SQLDBN=Syspro80db
SQLADM=Pb/cbyY4bWvQCek/wjFeofJEBwo7+jD0DB/Z2L
SQLSTD=KgHIaxPABjbRya+TEQzAw4sv1xXDqRC1kZyG+
SQLBLK=C:\Temp
SQLCPG=01252
```

**Note:** SQLADM= defines the administrative SQL username and password (encrypted) and  
SQLSTD= defines the standard SQL username and password (encrypted)

If either of the SQL login names are no longer valid or their SQL passwords are incorrect, lost or expired, you may need an alternative way of getting your SYSPRO system up-and-running again.

SYSPRO 8 provides a 'SQL authentication reset mode' allowing to gain access to the initial SQL login process. This is explained in more detail below.

## SETTING UP A STANDARD AND ADMINISTRATIVE SQL USER

Before using the 'reset' function you must ensure that you have two appropriately configured standard and administrative SQL users in SQL Server.

The administrative SQL user requires administrative rights to the SYSPRO system-wide and company specific databases. Typically requiring 'sysadmin' or appropriate CREATE DATABASE and BULK INSERT permissions.

The standard SQL user requires **db\_datareader** and **db\_datawriter** rights to the SYSPRO system-wide and company specific databases.

Once you have the details to these two SQL logins you are ready to initiate the 'SQL authentication reset mode'.

## INITIATING THE 'SQL AUTHENTICATION RESET MODE'

Save your IMPACT.INI before making any changes.

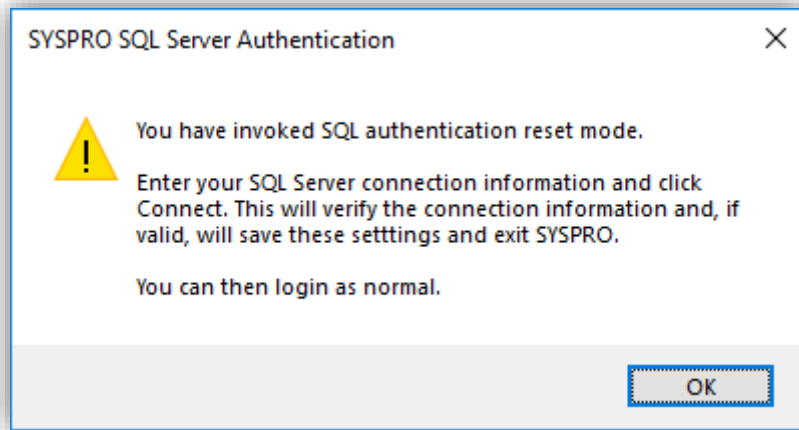
Edit IMPACT.INI using Notepad, or any other text editor, and change either the SQLADM= or SQLSTD= entry to the value '[reset]' (lower-case).

For example:

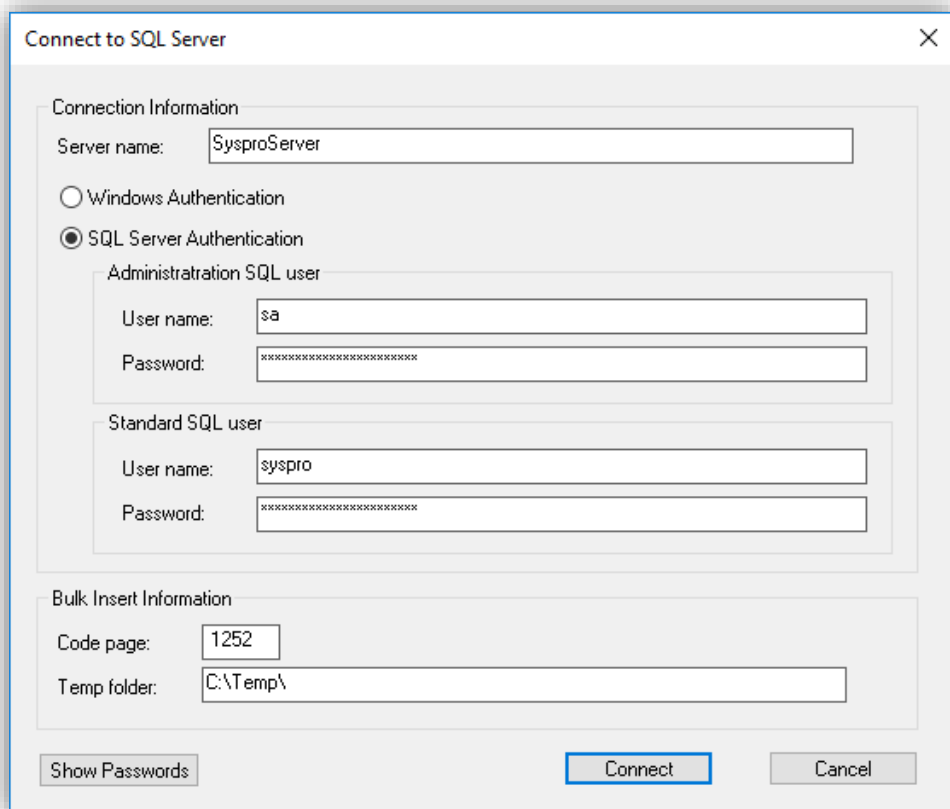
```
[Database Settings]
SQLLGN=SQLSERVER
SQLSSN=SysproServer
SQLDBN=Syspro80db
SQLADM=[reset]
SQLSTD=
SQLBLK=C:\Temp
SQLCPG=01252
```

Now login to SYSPRO as normal.

SYSPRO will enter a 'SQL Authentication reset mode' – you will be presented with the following confirmation:



Once you click 'OK' you will be prompted for the information required for SYSPRO to connect to SQL Server. See the following dialog:



You should then click 'Connect' – this will test the SQL connection information and if validated will update your `IMPACT.INI` and exit SYSPRO.

You are then ready to login to SYSPRO as normal.

# Insight into SYSPRO Applications and their Interaction with SQL Server

---

This set of topics provides insight into how SYSPRO applications interact with SQL Server.

This includes:

- Database access using a connection string
- Generic database access
- Optimized database access and diagnostics
- Transaction Processing
- Optimistic Concurrency Control and Timestamps

Whilst this document introduces these topics, developers will find significantly greater depth of information by using the Software Development Kit and related documentation.

## **SYSPRO USES A CONNECTION STRING TO CONNECT TO SQL SERVER**

As mentioned under the topic [Configuring SYSPRO to work with SQL Server](#), SYSPRO uses ODBC to connect to SQL Server as Microsoft has adopted ODBC as the de-facto standard for native access to SQL Server.

ODBC is a high-performance native API for connecting and communicating with SQL Server.

SYSPRO 8 uses a connection string to connect to SQL Server. This connection string is built automatically based on the SQL Server name and authentication methods configured in the **Setup Options** program (*Setup Options > System Setup > SQL*).

## **SYSPRO APPLICATIONS HAVE TWO DATABASE ACCESS METHODS AVAILABLE**

SYSPRO developers have two main sets of APIs that they can use to access SQL Server:

1. Generic database access logic
  - Useful for most common business logic
  - Used when inserting, updating, or deleting single rows
  - Used to retrieve rows sequentially when reading via the primary key or alternate indexes – can be used to update row by row with complex business logic if required.



## 2. Optimized database access logic

- Generally used when increased performance and scalability is critical
- Provides set-based operations  
*Updating and deleting multiple rows with single statements*
- Retrieving rows with complex WHERE conditions
- Retrieving rows from multiple tables using JOINS
- Customizing the database structure  
*For example, the Custom Form designer can add, change, or delete a column from the table holding the custom data.*

Developers often use the Generic database access for most of their business logic, however they can also use the Optimized database access when required. These two sets of database access logic can be used as required in virtually any relevant combination.

### GENERIC DATABASE ACCESS

The Generic database access logic has been available to SYSPRO developers for many years and provides a simple to code, robust and effective database access method. It is also designed to isolate many details of the underlying database from developers. The performance is very good for most business logic.

The Generic database access makes use of a set of dedicated 'SQL file handlers' – one per table. These file handlers are statically built once 'per release' of SYSPRO so that they exactly match the SYSPRO data dictionary. The file handlers are in the application server `BASE\FH` and `BASE\FH64` folders.

The SQL file handlers issue SQL statements that access all standard SYSPRO columns when selecting, inserting, or updating rows in a table. This is 'by design' and part of the simplification process that makes using the generic database access simple for developers. Also, for this reason, any User Columns are never referenced by the Generic database access.

If you use a SQL trace facility you may notice that all the standard columns are being accessed when data is being queried, updated, or inserted against a specific table. This typically indicates that Generic database access logic is being used.

Remember, most ERP transactions consist of singleton selects, updates and inserts – this lends itself to this generic database access logic.

### OPTIMIZED DATABASE ACCESS

The Optimized database access logic requires the developer to have a more thorough understanding of SQL Server, the SYSPRO database architecture, the SQL Server query optimizer (including hints) and other performance considerations. However, effective use of Optimized database access can provide significant performance benefits for some business logic.

SYSPRO applications that wish to verify or modify the SYSPRO database architecture require to use the Optimized database access. The Generic database access method does not provide a method of modifying the database architecture. For example, the Custom Form designer uses Optimized database access.

If you use a SQL trace facility, you may notice SQL commands that access a subset of the columns in a table, commands that access multiple tables with relevant JOINS, more sophisticated WHERE logic or even set-based updates and deletes. This indicates that Optimized database access logic is being used.

## OPTIMIZED DATABASE ACCESS - DIAGNOSTICS

Developers, Testers, Support personnel and others may find it useful to view the Optimized database access statements being issued by SYSPRO applications.

You can view the Optimized database access statements being issued against SQL Server by defining the **Diagnostics** option as 'sql02' in the **Setup Options** program (*Setup Options > System Setup > General*). See example screen shot as follows:

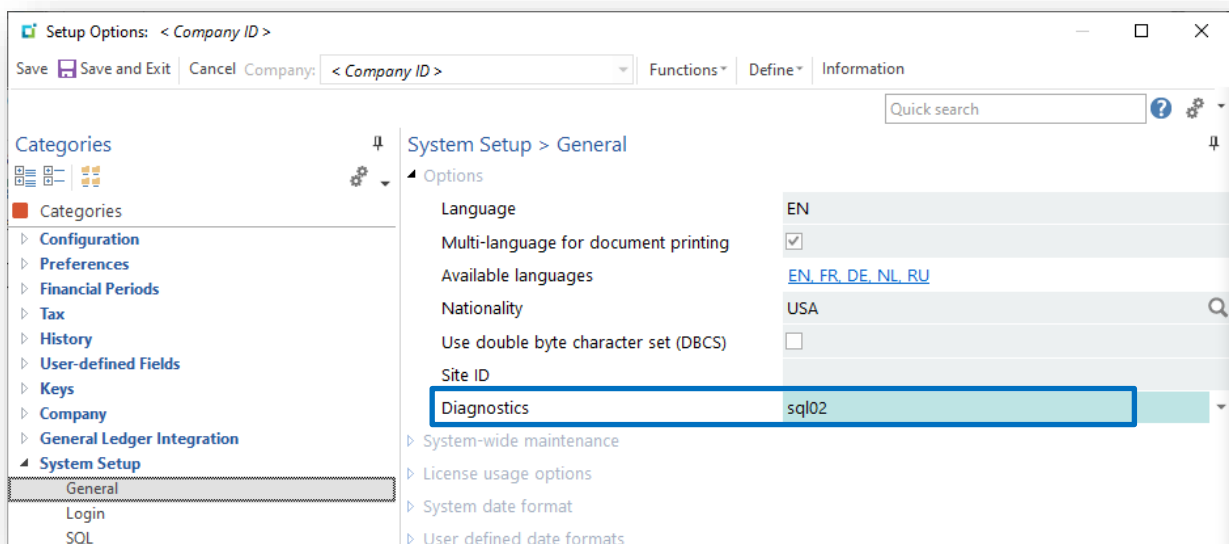


Figure 9: **Setup Options** > *System Setup > General > Options > Diagnostics*

When configured, all Optimized database access SQL statements are logged to a file on the application server named:

```
BASE\SETTINGS\ADMIN.SQL
```

Where 'ADMIN' is the name of the operator.

**Warning:** You should only use this diagnostic on a development or test environment (or for very short periods in a live environment) as the diagnostic files add an overhead to the database access and can grow very quickly.

Also note that this diagnostic only logs Optimized database access - the Generic database access is NOT logged using this diagnostic. Therefore, you should be aware when running a SYSPRO application that you may not be seeing the entire SQL Server interaction.

The 'sql02' diagnostic also includes information about each time a SYSPRO application connects and disconnects from the administration account configured in the **Setup Options** program.

For example, when you run the SQL Server Diagnostic program ('IMPDG6') you could see lines in the diagnostic log as shown below (in this case I just loaded and exited the diagnostic program).

```
-- Command generated 2018/10/24 17:15:20 Program: IMPSLG Called from: IMPDG6
Connected as user: sa

-- Command generated 2018/10/24 17:15:20 Program: IMPSLG Called from: IMPDG6
Issued: USE Syspro80db

-- Command generated 2018/10/24 17:15:21 Program: IMPSLG Called from: IMPDG6
Disconnected from current connection
```

The Report Writer also appends to this log when running a report (and when the 'sql02' diagnostic is enabled).

## SQL SERVER PROFILER

Sometimes when trying to diagnose performance or locking/blocking issues, support personnel or database administrators may run the SQL Server Profiler. This is an interface to create and manage traces and analyze trace results.

The subject of configuring, setting up and/or replaying using the SQL Server Profiler utility is not covered in this document.

We have had several questions about the output of the trace. Some of which will be answered here.

I have repeated a part of the introduction about OLTP systems as it does have some relevance to the actual SQL statements being issued in SYSPRO applications (and these are visible in a trace created by the SQL Server Profiler).

The following is an extract from a Microsoft Technical Reference Guide on OLTP systems (<http://technet.microsoft.com/en-us/library/hh393556.aspx>):

*“Operational, or online transaction processing (OLTP), workloads are characterized by small, interactive transactions that generally require sub-second response times. It is common for OLTP systems to have high concurrency requirements, with a read/write ratio ranging from 60/40 to as low as 98/2. Modifications are predominantly singleton statements, and most queries are constrained to simple joins. While limiting joins to as few tables as possible is desirable, a significant number of application systems do join many tables...”*

## SQL SERVER PROFILER – CURSOR VS SET BASED OPERATIONS

As mentioned in the fragment included above, the majority of SYSPRO's interaction with SQL Server is small, interactive queries and/or transactions, both requiring quick response. Overall, it is far more likely that a SYSPRO application will read data than write or change data.

Also, as mentioned earlier, SYSPRO applications have two sets of methods (APIs) to access SQL Server – these are known internally as 'generic' and 'optimized' access. It was also mentioned how you can distinguish between which method is being used.

When viewing a SQL trace, you may notice that simple procedures / batches are executed when viewing data – the results are fetched via a cursor. Depending on the application requirements it may access a single row (singleton SELECT) or, when required, multiple rows may be fetched and returned to the application. The rows to be fetched are selected using an appropriate WHERE clause and may optionally be sourced from multiple tables via an appropriate JOIN.

In some cases, complex business logic must be applied to each row as it is processed. The application may do this by fetching each row, performing the complex business logic (including sometimes inserting or changing rows in the same or a different table) and often updating the row to complete the logic – for example, by decrementing a value or setting a status. The business logic is complex enough that a set-based operation is not suitable or desirable. Also, this processing often encompasses just a few rows but can scale when required. When performed inside a transaction appropriate locking – Update Intent locks and Exclusive locks will be used to ensure transaction integrity.

Examples include creating an invoice from a sale order. Each line must be processed, and different logic applied based on the order line type. Merchandised (stocked) items are handled differently to non-stocked lines, freight, miscellaneous charges, and other line types. Stocked lines may require complex business logic to process associated lots, bins and/or serial numbers. All of this is further complicated by line and/or order discounting and sophisticated pricing policies together with local tax regulations (i.e. it's not a simple set-based operation).

Where possible, applications often do issue set-based operations – for example, setting a flag for all items that match a specific condition; or deleting all rows that are obsolete and are ready to be purged, etc.

## TRANSACTION PROCESSING

Data integrity is one of the most significant reasons for using SQL Server to store and transact data. When correctly configured (and when appropriate backup models are in use) SYSPRO and SQL Server provide enterprise strength data integrity. SYSPRO applications use Transaction Processing concepts to help guarantee this data integrity.

This topic briefly introduces the subject of Transaction Processing and SYSPRO's use of this technology.

## WHAT IS A TRANSACTION?

A Transaction is a set of business logic that affects the database. It is a logical unit of database operations which are executed.

Transactions should have four properties:

- Atomic
- Consistent
- Isolated
- Durable

These properties of database transactions are often referred to by the acronym ACID.

**Atomicity:** A transaction is an indivisible unit. Either all its data modifications are performed, or none of them are performed.

**Consistency:** Transactions must keep the database from one consistent state to another consistent state. Consistency is closely related to atomicity.

**Isolation:** A transaction is implemented and cannot be interfered with by other transactions. That is, an internal operation of a transaction and the use of the data is isolated from other transactions, the concurrent implementation of all transactions cannot interfere with each other.

**Durability:** Also called permanence. It refers to a transaction which is submitted, and the data in the database it changes should be permanent. The next operation, or other faults, should not have any impact on them and cannot be undone. The modifications persist even in the event of a system failure.

*For more information see: <http://msdn.microsoft.com/en-us/library/aa480356.aspx>*

The classic example of a Transaction in the SYSPRO ERP product is where the General Ledger has a debit posted to one account, in which case there must immediately be an equivalent credit posted to a different account, ensuring that the ledger is 'in balance'.

In this case, the Transaction must consist of both database changes (the debit and the credit) or it must have neither change. What you must never have is a debit without a credit or vice versa.

## TRANSACTION PROCESSING IN SYSPRO APPLICATIONS

Transaction Processing is implemented in SYSPRO applications by issuing a SQL Server BEGIN TRANSACTION statement before the transaction starts, and a COMMIT TRANSACTION after the transaction has made its last database change.

If any unexpected problem occurs between the BEGIN and COMMIT, then a ROLLBACK TRANSACTION is issued. This tells SQL Server to undo all database changes (and releases any locks held) so that the database is in the same state as it was before the BEGIN TRANSACTION statement.

In most SYSPRO applications, if they encounter an 'unexpected problem' during a transaction, they will show an appropriate message to the operator. Note that when the message has been shown the transaction has already been rolled back (undone).

See the following sample message (this specific message was generated for demonstration purposes only):

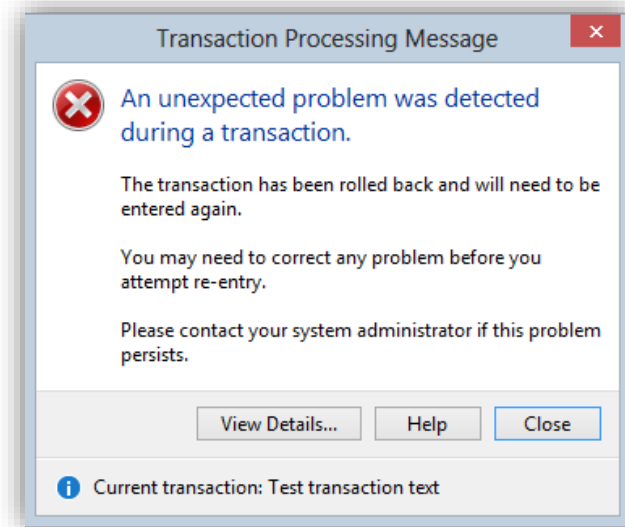


Figure 10: Sample Transaction Failure Message

When this message has been displayed, the operator can click 'View Details...' and then 'Extended SQL Information...' to receive more information about the cause of the 'unexpected problem'.

See the example below:

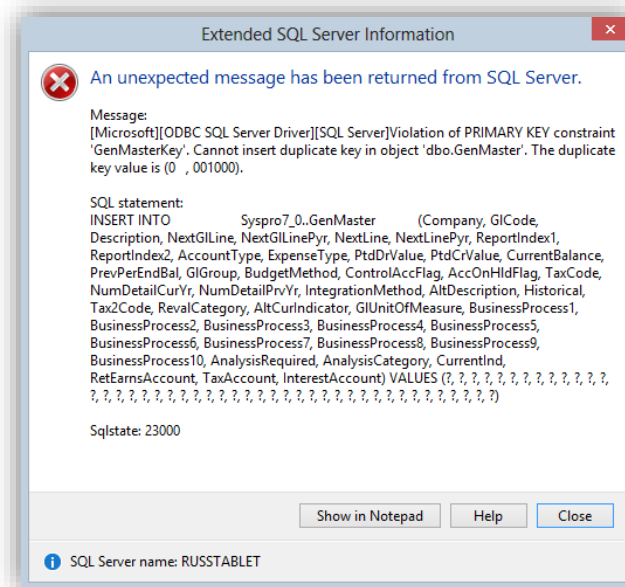


Figure 11: Example Extended SQL Information

In this case the SYSPRO application attempted to insert a duplicate primary key – you can see that by the message: *Violation of PRIMARY KEY constraint 'GenMasterKey'*.

Database access performed between a BEGIN and its associated COMMIT causes locks to be acquired for each row being affected. If another application attempts to access any of these (locked) rows they will be 'blocked' (i.e. they will wait until the Transaction completes either via a COMMIT or ROLLBACK).

SYSPRO applications are designed to ensure that the database access performed between BEGIN and COMMIT occur as 'quickly as possible' to reduce contention and resource usage. This helps prevent excessive blocking (slowing of other applications as they wait for your transaction to complete).

SYSPRO applications never interact with the user while a Transaction is in progress as, depending on the speed of the operator's response, there could be excessive blocking.

It was mentioned earlier that 'most SYSPRO applications' show the messages discussed above. The alternative is when business objects are run in an e.net environment. In this case the transaction is also rolled back – however instead of a message 'being displayed to the user', an exception message is returned to the calling application indicating that the rollback has occurred.

## TRANSACTION PROCESSING AND JOB LOGGING

In the event of a transaction being rolled back due to an 'unexpected problem', a transaction status of 'Rolled back' will be inserted into the Job Logging system.

It's possible to filter on this transaction condition so that you can see whether an excessive number of rollbacks are occurring. See the following example Job Logging Query where a transaction selection of 'Rolled back' has been selected:

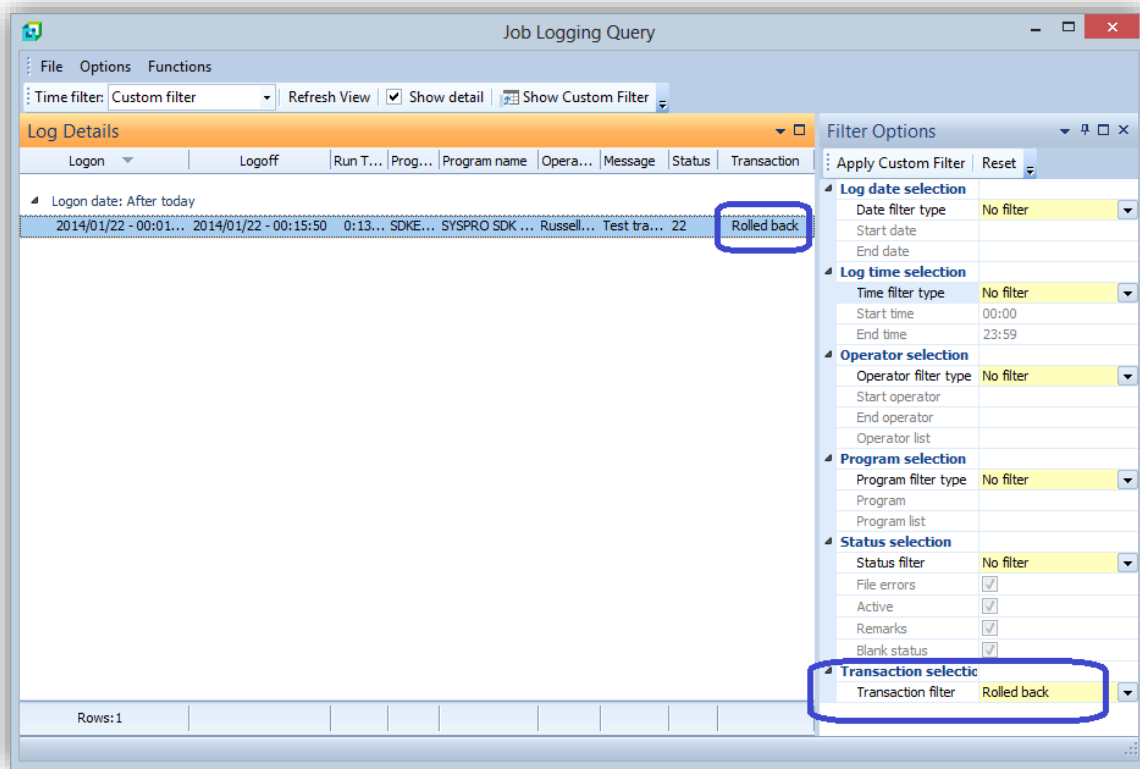


Figure 12: Job Logging - Showing Rolled Back Transactions

In the event of an excessive number of Rolled-Back transactions, you should investigate the cause further.

If the Rollbacks are occurring from a single SYSPRO application, you should consider viewing the 'Program version' by adding this column to the list view using the Field Chooser function. This may help you to identify whether this is a known issue and requires a software fix, or whether there is some other reason for the contention.

## TRANSACTION PROCESSING AND DEADLOCKS

Occasionally you may experience deadlock problems. These can be difficult to understand and consequently solve. This topic is designed to explain a deadlock and what occurs when a deadlock is detected.

### **A deadlock can be described with the following example:**

Suppose you have two rows in a table – item 'A' and item 'B'. Also suppose there are two operators using SYSPRO. The first operator starts a transaction locking item 'A' and the second operator starts a transaction and locks item 'B'.

Now the first operator (whilst still inside the same transaction) attempts to lock item 'B'. The first operator will be 'blocked' as item 'B' is locked by the second operator. This is to be expected and correct. Typically, the second operator will complete their transaction, thus committing their changes and releasing the lock on item 'B' – in which case the first operator's request to access item 'B' is satisfied.

However, suppose that the second operator now attempts to access item 'A'. Now we have a deadlock (sometimes also known as a 'deadly embrace'). Each operator has locked one item and are attempting to access the item locked by the other operator. If both block indefinitely, then both transactions will appear to 'hang'.

This can be more serious than just these two transactions not completing. In a real-world situation there are often other transactions that also start to be blocked as they wait for either of the first two to complete. This could escalate until a significant proportion of database users are unable to complete any Transactions.

### **Deadlock Detection**

The SQL Server Database Engine has a deadlock detection scheme. Deadlock detection is performed by a lock monitor thread that periodically initiates a search through all the tasks in an instance of the Database Engine. Initially this is initiated approximately every 5 seconds. Once it detects a deadlock it continues to monitor deadlocks more frequently until it determines it can revert to the original time increment.

*After a deadlock is detected, the Database Engine ends a deadlock by choosing one of the threads as a deadlock victim. The Database Engine terminates the current batch being executed for the thread, rolls back the transaction of the deadlock victim, and returns a 1205 error to the application.*



*Rolling back the transaction for the deadlock victim releases all locks held by the transaction. This allows the transactions of the other threads to become unblocked and continue.*

For more information see: [http://technet.microsoft.com/en-us/library/ms178104\(v=sql.105\).aspx](http://technet.microsoft.com/en-us/library/ms178104(v=sql.105).aspx)

Due to the above Deadlock Detection mechanism, you should not have the situation where a significant proportion of SYSPRO users are blocked, preventing them from continuing with their transactions. However, you may find that an 'excessive' number of users are becoming the 'victim of a deadlock'. If this occurs, you should investigate the cause as described above against – 'Transaction Processing and Job Logging'.

**Note:** The [User Triggers](#) topic earlier in this document provides some useful guidelines when a User Trigger appears to be causing an excessive number of Deadlocks or Rollbacks. We have found that this is often the most common cause of Deadlocks or Rollbacks on a SYSPRO database.

## OPTIMISTIC CONCURRENCY CONTROL AND TIMESTAMPS

OLTP systems like SYSPRO rely heavily on Transaction Processing to ensure data integrity.

Transaction Processing uses locks to ensure transaction isolation. With this comes the issues of blocking (and potentially deadlocks) as described earlier.

However, in SYSPRO there are many real-world situations where it is extremely unlikely that two or more operators will attempt to update the same information at the same time.

For example: 'AR invoice terms' provides a lookup table with a description, discount percentage and number of days within which an invoice discount may apply. In most companies these agreed AR invoice terms are subject to contracts and only change rarely. Therefore, it is extremely unlikely that two or more operators would use the AR invoice terms maintenance program to change a specific AR invoice terms code at the same time.

### INTRODUCING OPTIMISTIC CONCURRENCY CONTROL


SYSPRO supports an alternative technique of ensuring data integrity where it is unlikely that two or more operators are changing the same item at the same time.

The concept is known as Optimistic Concurrency Control and uses a special database capability to manage something called a Timestamp.

### TIMESTAMPS

A Timestamp is a database-wide value that is incremented each time a row is inserted or changed in any table. If a table has a Timestamp column then the timestamp value is automatically updated by the database each time a row is updated or inserted.

All SYSPRO tables contain a Timestamp column for this purpose.



**Important:** In this context, a Timestamp is not a date or time but is an integer that is incremented by the database engine each time a row is updated or inserted in the database. At any time, the database stores the current timestamp integer value. Sometimes a timestamp data type is also known as a 'rowversion'.

The following steps demonstrate how a Timestamp, together with appropriate code in a SYSPRO application, provides Optimistic Concurrency Control.

- Initially an operator starts to edit a row in a table.  
When the row is selected, the application records the timestamp value stored against the row. For this example, assume a timestamp value of 25.
  - Note there is no lock on the row.
- The operator then changes one or more values using the SYSPRO application user interface.
  - Again, note that there is no lock on the row.
  - The operator can work with the user interface and take as long as they like.
- Eventually, sometime later, the operator clicks 'Save'. Now, just before the row is updated, the SYSPRO application re-reads the timestamp value and if it is still the same (25) then the row is updated.
  - As the timestamp value has not changed, the database has guaranteed that no one else has changed any value against the row since the row was first selected for editing.
- However, if the timestamp value has changed (for example has a value of 26) then the row is not updated, and a Timestamp Mismatch Message is displayed to the operator indicating that someone else changed the row unexpectedly.
  - When the timestamp value has changed it indicates that some other process has made one or more changes to the row since the initial operator started the editing process. This could be a SYSPRO application (or any application) as it's the database that manages the Timestamp column and its value.
  - In most cases, once the Timestamp Mismatch Message is displayed, the operator must re-start editing the item. The SYSPRO applications force a restart to the data entry process, as it is expected that this is such a rare occurrence that there is no further logic required.
  - An example Timestamp Mismatch Message is shown below.
- If the operator does not 'Save' any changes but instead cancels out of the maintenance, then the program simply returns to the menu (or from where it was invoked).
  - As no lock was acquired, there are no resources to unlock or clear.
  - Similarly, if the client application fails or is terminated, or the server-side application fails or is terminated, there was no lock acquired and therefore there is nothing to undo.
  - See the following sample Timestamp Mismatch Message (this specific message was generated for demonstration purposes only):

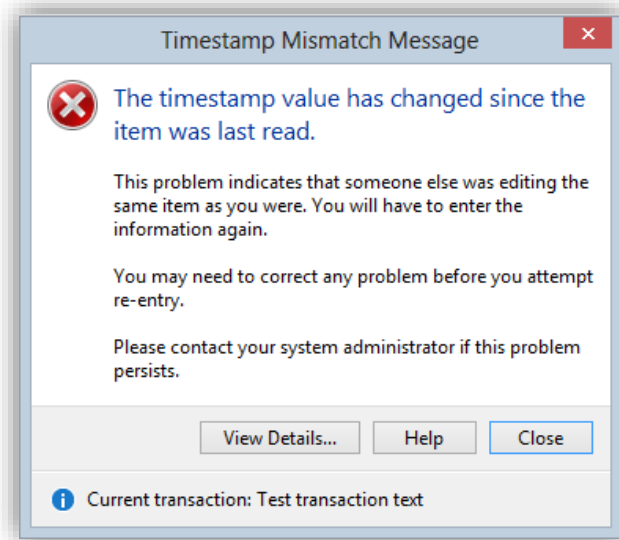


Figure 13: Sample Timestamp Mismatch Message

The maintenance scenario described above applies to many of the simple maintenance applications in SYSPRO.

The advantages of using Optimistic Concurrency Control include:

- There are no locks involved and therefore there is no blocking and no chance of deadlocks.
- As there are no locks there are fewer resources used – improving performance and scalability.
- As Timestamps are managed by SQL Server, this logic applies even if a third party attempts to change rows when an operator is using a SYSPRO application to change the same item.
- Similarly, third parties could use the same technique when changing data on SYSPRO tables.
- There are many simple maintenance applications where this logic works perfectly in a real-world business environment.

However, Optimistic Concurrency Control **cannot** be used in the following environments:

- Transactions  
Most Transactions require two or more rows to be updated in the same or different tables. *Remember the classic Debit and Credit example.*
  - The reason that timestamps cannot be used in this instance is that the application would have to remember a timestamp for each affected row. Then, when a database change was about to be made on the first row, the timestamp may not have changed and therefore the application would make the first update. The next row to be updated may detect a different timestamp from the originally saved timestamp and it would have to 'cancel' the process. However, the first update was performed. This means that there is no guarantee that all parts of the Transaction have been performed and therefore data integrity is not guaranteed.

- Similarly, if the SQL Server software, the server on which it is running or one of the physical disk drives crashes, then there is also no guarantee of data integrity.
- Note that when using Transaction Processing on a correctly configured SQL Server environment using the BEGIN and COMMIT transaction logic, SQL Server can guarantee the Transaction data integrity – even in the event of a system failure.
- Maintenance where multiple rows are affected
  - For the same reasons as the previous point, if a maintenance program requires to update two or more rows (in the same or different tables) then the same issues arise.
  - Due to this reason only relatively simple maintenance programs are suitable for Optimistic Concurrency Control using Timestamps.

## OTHER USES OF TIMESTAMPS – SYSPRO ANALYTICS

Timestamps can be used for purposes other than for Optimistic Concurrency Control.

SYSPRO Analytics takes advantage of another use of Timestamps and is briefly described below.

Each time the SYSPRO Analytics ‘extract job’ is performed, it requires extracting data from the SYSPRO database to a ‘staging’ database. This is virtually an ‘exact copy’ of the standard SYSPRO database.

It’s important that the extract process is as quick and efficient as possible. The first time that the extract process is run, SYSPRO Analytics takes a copy of each SYSPRO table. Then it identifies and stores the highest timestamp found.

The next time the extract process runs it only copies rows from each table where the timestamp against the row is higher than the ‘previously saved highest timestamp’. Once the extract is completed it then saves the new highest timestamp, ready for the next extract.

This technique ensures that only new rows, and rows that have been updated since the last extract, are copied to the staging database.

This is a fast and efficient mechanism to minimize the amount of data copied.

# SYSPRO and SQL Server Data Encryption

---

## INTRODUCTION

This topic focuses on data encryption relating to SYSPRO and SQL Server. It should be considered as part of your company's overall security and privacy policies.

The focus is on securing data from the SYSPRO ERP application when using Microsoft SQL Server and securing the communication *between* SYSPRO and SQL Server.

**Note:** The remainder of this document topic includes some content from external sources. For clarity, we have highlighted the relevant content in *italicized blue text* preceded by the original URL. As these web sites are out of our control, the text may have changed subsequently, or the URL may no longer be valid.

## AUDIENCE

This topic provides an overview of two technologies known as **Data Encryption at Rest (TDE)** and **Data Encryption in Motion (TLS)**.

If you require a more comprehensive overview, or a more detailed explanation about how to configure SYSPRO, SQL Server and Windows to work with these technologies, then see the SYSPRO Online Help Resources > Technical guides: **SYSPRO and SQL Server Encryption Overview** and its companion document: **SYSPRO and SQL Server Encryption Configuration**.

## WHY DATA ENCRYPTION?

In today's highly regulated world, most companies operate in an environment where they must comply with one or more security and/or privacy regulations or government acts. Examples include:

- EU citizens (GDPR - General Data Protection Regulation)
- Australia (OAIC - Privacy Act)
- Canada (PIPA, PIPEDA - Privacy Act)
- South Africa (POPI - Protection of Personal Information Act)
- USA (When this topic was created (September 2019) the United States did not have any centralized, formal legislation at the federal level regarding this issue. However, it does ensure the privacy and protection of data through the United States Privacy Act, the Safe Harbor Act and the Health Insurance Portability and Accountability Act. Individual states may also have relevant acts that apply).

Failure to comply with these regulations can often incur heavy penalties and even criminal prosecution.

In the event of a data breach, there are typically prescribed reporting considerations to a regulatory body.

In most cases the penalties for a breach can be largely mitigated if it can be shown that reasonable attempts were taken to protect your data. For example, if you can show that you have encrypted the database then any network or other security breach will limit the damage and, consequently, penalties can be mitigated.

In addition, by encrypting data passed between SYSPRO and SQL Server you effectively remove the chance of eavesdroppers and hackers being able to gather or even change data.

Owing to these considerations, many companies should consider data encryption as part of their security and privacy data compliance strategy. The following topics will be introduced, together with some information relating to the available technologies and some performance considerations:

- Data Encryption at Rest
- Data Encryption in Motion

## DATA ENCRYPTION AT REST

**Data Encryption at Rest** describes the technique of configuring SQL Server so that the physical database files stored on the Windows file system are encrypted. This ensures that, in the event of a network or other security breach, even if someone can access the physical database data or log files (or a backup of these files) the information remains secure.

The technique described here is known as **TDE - Transparent Data Encryption**.

Extract from: <https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/transparent-data-encryption?view=sql-server-2017>

*Transparent Data Encryption (TDE) encrypts SQL Server, Azure SQL Database, and Azure SQL Data Warehouse data files, known as encrypting data at rest. You can take several precautions to help secure the database such as designing a secure system, encrypting confidential assets, and building a firewall around the database servers. However, in a scenario where the physical media (such as drives or backup tapes) are stolen, a malicious party can just restore or attach the database and browse the data. One solution is to encrypt the sensitive data in the database and protect the keys that are used to encrypt the data with a certificate. This prevents anyone without the keys from using the data, but this kind of protection must be planned in advance.*

*TDE performs real-time I/O encryption and decryption of the data and log files. The encryption uses a database encryption key (DEK), which is stored in the database boot record for availability during recovery. The DEK is a symmetric key secured by using a certificate stored in the master database of the server or an asymmetric key protected by an EKM module. TDE protects data "at rest", meaning the data and log files. It provides the ability to comply with many laws, regulations, and guidelines established in various industries. This enables software developers to encrypt data by using AES and 3DES encryption algorithms without changing existing applications.*

## RELEVANT SYSPRO VERSION – ALL VERSIONS

Data Encryption at Rest using TDE requires SQL Server configuration and does NOT require any additional configuration from within the SYSPRO application.

Therefore, Data Encryption at Rest using TDE is applicable to any SYSPRO version.

**Note:** SYSPRO software prior to SYSPRO 8 stores some configuration and auditing data in the file system, and therefore is not encrypted using this technique. This is one of the many benefits of organizations migrating from earlier versions to SYSPRO 8.

The technical details of how-to setup and configure SQL Server with TDE are available in a separate technical guide available from within the [SYSPRO 8 Online Help](#) (see: *SYSPRO Help > Resources > Technical Guides > SYSPRO and SQL Server Encryption Configuration*).

## PERFORMANCE RUNNING SYSPRO AND SQL SERVER USING TDE

The SYSPRO development team have a set of benchmarks to test the software under load. Although these are run in a Microsoft Azure environment, an equivalently configured on premise environment would also be expected to yield the same results.

During 2019, the SYSPRO development team ran the benchmarking application using SYSPRO 8 2019 R2 software connecting to SQL Server 2017 with TDE enabled. Results were compared with the same environment, the same SYSPRO software and the same database not using TDE (i.e. a direct comparison between encrypted and unencrypted databases).

All the tests ran successfully, even under load, and the comparison showed that the average performance throughput was only degraded by about 1%.

## DATA ENCRYPTION IN MOTION

**Data Encryption in Motion** describes the technique of configuring SYSPRO and SQL Server so that all communication between SYSPRO and SQL is encrypted. This includes initial connection information, SQL statements issued, and the actual data passed to-and-from SQL Server.

Data Encryption in Motion ensures that eavesdroppers and hackers can't see what is transmitted. This is particularly useful for private and sensitive information, but also for all information sent between SYSPRO and SQL Server.

It should be mentioned that if the SYSPRO Application server and SQL Server are running on the same server, then Data Encryption in Motion may add an unnecessary overhead with little or no benefit.

The technology described here is known as **TLS – Transport Layer Security**.

Extract from: <https://blog.coeo.com/securing-connections-to-sql-server-with-tls>

*Fundamentally, TLS provides you with the ability to encrypt connections between SQL Server and calling client applications. When a client requests an encrypted connection to a SQL Server configured for TLS, an initial handshake takes place to negotiate the cipher suite from which further communication should take place. Once agreed, SQL Server then sends its TLS certificate to the client, which the client must then validate and trust against its copy of the Certification Authority (CA) certificate. Finally, providing the TLS certificate is trusted and it meets certain other requirements, a secure connection is established.*

**Warning:** You must use TLS 1.2 (or higher) as earlier versions had known vulnerabilities. For your information, TLS supersedes its now deprecated predecessor – SSL - Secure Sockets Layer.

## RELEVANT SYSPRO VERSION – SYSPRO 8 2020 R1

SYSPRO 8 2020 R1 (release February 2020) has been enhanced to allow an administrator to configure SYSPRO and SQL Server using TLS, thus providing Data Encryption in Motion.

### ODBC DRIVER INFORMATION

SYSPRO communicates with SQL Server using ODBC drivers provided by Microsoft. These provide standardized, robust, and high-performance interfaces to SQL Server.

SYSPRO 8 2020 R1 supports three different ODBC Drivers:

- SQL Server
- ODBC Driver 13 for SQL Server
- ODBC Driver 17 for SQL Server

The first driver (simply named **SQL Server**) ships as part of Windows and is known as a Windows Data Access Component (WDAC) – it has provided ODBC access to SQL Server for applications such as SYSPRO for many years. However, Microsoft have recently indicated that new software should no longer use this driver, partly because some features (e.g. relating to encryption) are not fully available with the **SQL Server** (WDAC) driver.

For this reason, SYSPRO 8 2020 R1 has been enhanced to allow more recent ODBC drivers to be specified in the **Setup Options** program (*Setup Options > System Setup > SQL*).

The remainder of this topic assumes that **ODBC Driver 17 for SQL Server** is selected. If you use the older driver (ODBC Driver 13 for SQL Server) then substitute the driver name when appropriate.

**Note:** You may have to download and install the required ODBC driver if it is not currently installed on your SYSPRO application server.



## ENCRYPTED CONNECTION STRING

Once you have chosen **ODBC Driver 17 for SQL Server** in the **Setup Options** program, you can select the **Encrypted connection string** checkbox and indicate whether you want a **Self-signed server certificate**.

These options are available in the *Setup Options > System Setup > SQL* form:

The screenshot shows a configuration window titled "System Setup > SQL". Under the "Company database connection" section, the following settings are visible:

- Company database authentication: SQL authentication
- SQL Server name: SysproServer
- SQL Driver to use: ODBC Driver 17 for SQL Server (highlighted with a blue box)
- Encrypted connection string:
- Self-signed server certificate:

**Warning:** Self-signed server certificates are, by their nature, less secure. The encrypted handshake is based on NT LAN Manager (NTLM). It is recommended that you provision a verifiable certificate on SQL Server for secure connectivity. Transport Layer Security (TLS) can only be secured with certificate validation.

Once you have saved these settings you should immediately configure SQL Server to work with TLS security. Alternatively, if you had already set up SQL Server to support TLS security then just login and the new Encryption in Motion technology is applied.

The technical details of how to set up and configure SQL Server with TLS is available in a separate technical guide available from within the [SYSPRO 8 Online Help](#) (see: *SYSPRO Help > Resources > Technical Guides > SYSPRO and SQL Server Encryption Configuration*).

## PERFORMANCE RUNNING SYSPRO AND SQL SERVER USING TLS

The SYSPRO development team have a set of benchmarks to test the software under load. Although these are run in a Microsoft Azure environment, an equivalently configured on premise environment would also be expected to yield the same results.

During 2019, the SYSPRO development team ran the benchmarking application using SYSPRO 8 2020 R1 software connecting to SQL Server 2017 with TLS enabled. Results were compared with the same environment, the same SYSPRO software and the same database not using TLS (i.e. a direct comparison between encrypted and unencrypted communication between SYSPRO and SQL Server).

All the tests ran successfully, even under load, and the comparison showed that the average performance throughput was only degraded by about 1%.

# SQL Health Dashboard

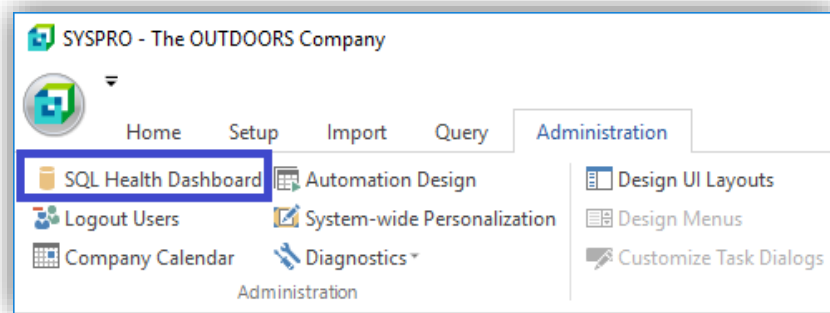
SYSPRO 8 includes a **SQL Health Dashboard** that provides a single place to monitor the key issues related to SYSPRO running on SQL Server.

It should be mentioned that there is no substitute for an expert system administrator and/or database administrator and/or SYSPRO support personnel when it comes to configuring, monitoring, diagnosing and correcting issues related to SYSPRO and SQL Server.

However, the **SQL Health Dashboard** is intended to provide some key information at-a-glance. The information shown, together with any highlighted items, is based on knowledge of our own software and how it interacts with SQL Server, together with input from our partners and customers about real-world issues that have been determined to have the most impact.

The SQL Health Dashboard is available from the ribbon bar:

**Administration > SQL Health Dashboard**



When you run the **SQL Health Dashboard** program, it analyzes your system-wide database and each company specific database. For larger number of databases this could take a minute or so.

Once the analysis has completed the dashboard will be displayed - see the following example:

The image shows a screenshot of the 'SQL Health Dashboard' application window. On the left, there is a table listing databases for different companies:

Company	Name	Database	Version
System	System database	Syspro8odb	8.0.0.0009
EDU1	The OUTDOORS Company	Syspro80Edu1	8.0.0.0009
EDU2	The OUTDOORS Company	Syspro80Edu2	8.0.0.0009
EDU3	Demo for Russell	Syspro80Edu3	8.0.0.0009

The main area of the dashboard shows details for the selected 'EDU1' database. It includes a 'Repair Issues' section with links for Tables (0 issues, 16 user defined), Columns (0 issues, 0 user defined), Indexes (0 issues, 0 user defined), Foreign keys (0 issues, 0 user defined), Object dependencies (0 dependencies), Index fragmentation (482 fragmented), and SQL users (6 users). Below this is a table of database files:

Logical Name	Type	Location	Size (MB)	Growth	Maximum (MB)	Used (MB)	Free (MB)	Free %
DS001_CMP_EDU1	Data	C:\SQL\Syspro80Edu1.mdf	500	10 %	Unlimited	333	167	33.40
DS001_CMP_EDU...	Log	C:\SQL\Syspro80Edu1_Log.ldf	121	10 %	2097152	5	116	95.87

At the bottom, there is a table showing database statistics and fragmentation:

Table	Index	Fragmentation %	Page Count	Rows in Table	Statistics Updated	Suggeste...
GenJournalDetail	GenJournalDetailKey	99.67	301	12732	2014/04/16 14:48:25.610	Rebuild
InvJournalDet	InvJournalDetKey	99.02	102	1966	2013/07/29 21:01:58.083	Rebuild
WipCapacityDyn	WipCapacityDynKey	95.24	21	3352	2013/10/17 13:19:32.500	Rebuild
AssetDistReg	AssetDistRegKey	92.86	14	2332	2014/04/16 14:41:37.363	Rebuild
lopAmendmentJnl	lopAmendmentJnlKey	92.31	13	213	2014/04/16 14:45:23.170	Rebuild
PorHistReceipt	PorHistReceiptIdxPord	91.30	23	3504	2013/07/09 20:55:09.013	Rebuild

You can now select one of the databases shown in the left pane – this displays the system-wide database together with each company specific database.

When you select one of the databases by clicking on the row, Database Details about the database are displayed on the top right and a series of tab pages are available - bottom right. You can either click on the one of the hyperlinks shown in the top right pane or click directly on each tab page to see the relevant information.

The following topics provide additional insight to each of the types of information related to the currently selected database.

## DATABASE DETAILS

The Database Details pane (shown top right) contains many properties and statistics relating to the currently selected database.

Highlights include:

- **Database version**

You should ensure that the database version matches the SYSPRO version as described in the topic: [Database Versions](#).

If the SQL health Dashboard finds any database that does not match the current database version it will not attempt to validate tables, columns, indexes, or foreign keys.

- **Collation**

You should ensure that the database collation conforms to the requirements of all SYSPRO databases – see the topic: [Collation and Case Sensitivity](#).

- **Compatibility Level**

You should ensure that the database compatibility level is set appropriately – see the topic: [SQL Server Compatibility Levels](#). If the compatibility level does not match you will see a warning symbol. If the warning icon is shown, please fix the compatibility level immediately.

- **Auto close**

SYSPRO databases must not have the 'Auto close' property enabled – see the following link: <https://docs.microsoft.com/en-us/sql/relational-databases/policy-based-management/set-the-auto-close-database-option-to-off?view=sql-server-2017>.

- **Auto shrink**

SYSPRO databases should not have the 'Auto shrink' property enabled as it can lead to performance degradation – see the following link: <https://support.microsoft.com/en-us/help/315512/considerations-for-the-autogrow-and-autoshrink-settings-in-sql-server>.

- **Status**

This shows the status of the database at the time the query was executed. If the status is in anything but ONLINE then the checks on tables, indexes, etc. are not carried out.

The database status can be:

- ONLINE
- RESTORING
- RECOVERING
- RECOVER\_PENDING
- SUSPECT
- EMERGENCY
- OFFLINE

See the following link: <https://docs.microsoft.com/en-us/sql/relational-databases/databases/database-states?view=sql-server-2017>.

- **Read only**

Live SYSPRO databases must never be set to 'read only'.

- **Last full backup**

Ensure that the date of last full backup is as expected for your planned backup regime.

- **Last log backup**

Ensure that the date of last full backup is as expected for your planned backup regime.

Database and log backup are outside the scope of this document.

See the following information regarding the database recovery model:

<https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/recovery-models-sql-server?view=sql-server-2017>

- **TDE encryption status**

This entry shows the current database encryption status. Use this to confirm whether the database is encrypted or not. For an overview of SQL encryption see the topic: [SYSPRO and SQL Server Data Encryption](#).

- **Change data capture enabled and change tracking enabled**

These options provide visibility to two methods of tracking database changes. Each has their own benefits and limitations and usually slow the database access. If you are having performance related issues you might want to verify these settings and investigate the requirements for them if enabled.

## DATABASE FILES

The Database File information is shown below the Database Details pane. This shows the following information for each file used to store the database master and log information:

- **Logical name**

Typically, the database name as you see it in SQL Server Management Studio and an associated log file.

- **Type**  
This is the type of file – it can be either Data or Log.
- **Location**  
The physical locations of data and transaction logs can have a significant effect on the system data integrity and performance. In a production environment these should be on separate drives for data integrity.
- **Size (MB)**  
This is the physical file size in megabytes in the file system.
- **Growth**  
This is the file growth property to be applied when free space is exhausted. It can be a percentage of the current file size or a number of megabytes.  
Setting this option too low can lead to excessive re-organization, disk fragmentation and poor performance. See the following link:  
[https://blogs.msdn.microsoft.com/john\\_daskalakis/2013/11/25/sql-server-default-settings-that-you-might-want-to-change/](https://blogs.msdn.microsoft.com/john_daskalakis/2013/11/25/sql-server-default-settings-that-you-might-want-to-change/).
- **Maximum (MB)**  
This is a configured maximum physical file size and can be a value in megabytes or 'unlimited'. On some versions of SQL Server, the default log can have a maximum value of 2097152 MB – this is just an 'arbitrary large' value and can be left 'as is'.
- **Used (MB)**  
Data usage in physical file.
- **Free (MB)**  
Free space in physical file.
- **Free %**  
This is the percentage of free space.  
This is the 'Free' space as a percentage of the file 'Size'.

The subject of setting up a new database, defining the file locations, size, free space, and growth factors is complex. This is discussed in the topic about [New Databases](#).

## SQL INSTANCE INFORMATION

There is an auto-hidden pane top right of the SQL Health Dashboard that shows information about the instance of Microsoft SQL Server. This is the same regardless of the database selected.

Highlights include:

- **SQL version**  
This shows the SQL Server version, including sub-versions and service packs when relevant.

- **Collation**

This is the collation defined when the SQL instance was installed and cannot be changed. It will be used for the system databases. It will govern the default collation. However, when you create SYSPRO databases you can override the collation when necessary.

- **Maximum server memory (MB)**

It is important in a correctly configured system to fix the maximum SQL Server memory preventing it from taking all available server memory. Failure to limit the SQL Server memory will often lead to a very poorly performing system, especially when SQL exists on the same server as the applications that use it.

See the following topic: <https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/server-memory-server-configuration-options?view=sql-server-2017>.

- **Connection time (seconds)**

The connection time provides an indication of how long SYSPRO takes to connect to SQL Server. It may not seem that important, as when you login to SYSPRO if the connection takes a hundredth of a second (1/100) or 1 second is a small issue once per login.

However, some of the SYSPRO code that has been optimized for SQL performance will issue SQL statements on a separate connection. If the separate SQL connection takes (say) 1 second, then each fragment of optimized code may in fact be slower than the code it was replacing. Thus, the connection speed does play an important part in the overall SYSPRO performance on SQL Server.

A value below 0.10 seconds is ideal and preferably 0.01 or 0.02 seconds for best performance. See the following topic: [Connection to SQL Server is Slow](#).

- **Connection driver**

This confirms the current ODBC driver used to connect to SQL Server.

See the topic: [Setup Options > System Setup > SQL: SQL Driver to use](#).

- **Connection encrypted**

This provides an indicator of the SQL connection encryption status. For an overview of SQL encryption see the topic: [SYSPRO and SQL Server Data Encryption](#).

- **Instant file initialization**


In Microsoft SQL Server, instant file initialization allows for fast execution of file operations and reclaims used disk space without filling that space with zeros. Instead, disk content is overwritten as new data is written to the files.

We recommend that you enable this option to ensure optimal performance for the growth of SQL tables on disk. A warning icon is displayed if it is not selected.

*This option is only applicable when using SQL Server 2016 and above.*

## SQL HEALTH DASHBOARD – TABLES

The Tables tab shows information about standard SYSPRO tables that are missing. These should be treated as serious errors and will prevent parts of the SYSPRO application from running.



In addition, any user-defined tables are also shown. These are tables that do not appear to be standard SYSPRO tables. This could be due to one of the following reasons:

- The table is not defined in our standard data dictionary.
- The table is not a custom form table (ends with a '+').
- The table is not a reporting archive table.

If the dashboard lists any user-defined tables then it is worth investigating what, or who, has created these tables and whether they are still required and/or relevant.

Do not just delete tables that appear in this list. Ensure you have thoroughly investigated before deciding to delete user-defined tables. Typically, user defined tables have no effect on the running of standard SYSPRO applications. However, they can affect the backup performance, disk consumption etc.

## REPAIR ISSUES

If there are missing tables, then the 'Repair Issues' toolbar button (top right) can be selected.

This will automatically add missing tables using the correct definition as per the data dictionary.

**Important:** If you find that one or more tables are listed as missing, are re-added when using the repair function, but later are listed as missing again, then you must investigate and prevent whatever (or whoever) is deleting these tables.

## SQL HEALTH DASHBOARD – COLUMNS

The Columns tab shows information about standard SYSPRO columns that are missing from standard tables – these should be treated as serious errors as they will prevent parts of the SYSPRO application from running.

Any standard SYSPRO column that does not conform to one-or more of the standard properties (such as datatype, length, collation, default, NULL indicator etc.) will be listed – these should be treated as serious errors and will prevent parts of the SYSPRO application from running.

In addition, any user-defined columns are also shown. These are columns in standard SYSPRO tables that do not appear to be standard SYSPRO columns. This includes user defined Computed Columns. This could be due to one of the following reasons:

- The column is not defined in our standard data dictionary
- The column is not a custom field defined using the custom form designer

If the dashboard lists any user-defined columns then it is worth investigating what, or who, has created these columns and whether they are still required and/or relevant.

Do not just delete columns that appear in this list. Ensure you have thoroughly investigated before deciding to delete user-defined columns. Typically, user defined columns have no effect on the running of standard SYSPRO applications.

## REPAIR ISSUES

If there are missing columns and/or columns with incorrect properties, then the 'Repair Issues' toolbar button (top right) can be selected. This will automatically add missing columns using the correct definition as per the data dictionary. It will also attempt to correct any incorrect properties – it is possible that inappropriate data may prevent the properties from being set to the standard values. Any errors will be shown and must be manually corrected.

**Important:** If you find that one or more columns are listed as missing, are re-added when using the repair function, but later are listed as missing again, then you must investigate and prevent whatever (or whoever) is deleting these columns.

If user-defined columns are not defined as nullable and have no default value assigned, then the repair will set these columns as nullable.

## SQL HEALTH DASHBOARD – INDEXES

The Indexes tab shows information about standard SYSPRO indexes that are missing from standard tables – these should be treated as serious errors and may degrade the performance of SYSPRO applications. Especially in tables with more than a few rows of data. In addition, if the primary clustered index is missing it may allow duplicates to be inserted into the table leading to data corruption and/or invalid data.

In addition, any user-defined indexes are also shown. These are indexes on standard SYSPRO tables that do not appear to be standard SYSPRO indexes. This could be due to one of the following reasons:

- The index is not defined in our standard data dictionary
- The index is not a primary clustered index on one of the custom form tables (ending with '+')

If the dashboard lists any user-defined indexes then it is worth investigating what, or who, has created these indexes and whether they are still required and/or relevant.

Do not just delete indexes that appear in this list. Ensure you have thoroughly investigated before deciding to delete user-defined indexes.

Please note that adding user defined indexes can have a negative effect on standard SYSPRO applications. See the topic: [Guidelines when adding User Indexes](#).

## REPAIR ISSUES

If there are missing indexes or any standard index has incorrect properties, then the 'Repair Issues' toolbar button (top right) can be selected.

This will automatically add missing indexes using the correct definition as per the data dictionary. It will also attempt to correct any incorrect properties – it is possible that inappropriate data may prevent the properties from being set to the standard values. Any errors will be shown and must be manually corrected.



**Important:** If you find that one or more indexes are listed as missing, are re-added when using the repair function, but later are listed as missing again, then you must investigate and prevent whatever (or whoever) is deleting these indexes.

## SQL HEALTH DASHBOARD – FOREIGN KEYS

The Foreign Keys tab shows information about standard SYSPRO foreign keys that are missing from standard tables – this should be treated as relatively minor information as no SYSPRO applications rely on foreign keys - all data integrity is completely managed using SYSPRO business logic.

Any standard SYSPRO foreign key that does not conform to one-or-more of the standard properties (such as the list and sequence of columns and the 'NOCHECK' property etc.) will be listed.

In addition, any user-defined foreign keys are also shown. These are foreign keys linking to/from standard SYSPRO tables that do not appear to be standard SYSPRO foreign keys. This could be due to one of the following reasons:

- The foreign key is not defined in our standard data dictionary

If the dashboard lists any user-defined foreign key then it is worth investigating what, or who, has created these foreign keys and whether they are still required and/or relevant.

Do not just delete foreign keys that appear in this list. Ensure you have thoroughly investigated before deciding to delete user-defined foreign keys.

### REPAIR ISSUES

If there are missing foreign keys, then the 'Repair Issues' toolbar button (top right) can be selected.

This will automatically add missing foreign keys using the correct definition as per the data dictionary.


**Important:** If you find that one or more foreign keys are listed as missing, are re-added when using the repair function, but later are listed as missing again, then you must investigate and prevent whatever (or whoever) is deleting these foreign keys.

## SQL HEALTH DASHBOARD – OBJECT DEPENDENCIES

The Object Dependencies tab shows information about links from third party objects into standard SYSPRO tables or columns. This includes custom form tables.

This can include:

- User Views
- User Stored Procedures
- User Computed Columns
- User Table and/or Scalar Functions
- User Triggers



If the dashboard lists any Object Dependencies then it is worth investigating what, or who, has created these dependencies and whether they are still required and/or relevant.

Do not just delete items that appear in this list. Ensure you have thoroughly investigated before deciding to delete user-defined objects or links from them to standard SYSPRO tables or columns.

**Important:** The column headed 'Updated' indicates that the user-defined object is updating a standard SYSPRO column.

We strongly advise against updating SYSPRO data directly, as any change may not follow the business rules associated with the data to be applied.

There are cases where custom form data may be updated directly – however you must ensure that any validation is honored.

## REPAIR ISSUES

There is no repair related to Object Dependencies as these are 'by definition' user defined.

## SQL HEALTH DASHBOARD – INDEX FRAGMENTATION

The Index Fragmentation tab shows information about the percentage fragmentation of each standard SYSPRO index. Only indexes where the fragmentation percentage is over 5%, or the index statistics have not been updated for at least 30 days, are included.

The list of indexes is shown in 'index fragmentation percentage' sequence with the highest fragmentation first.

General guidelines say that index fragmentation percentages above 30% can lead to performance degradation. However, it is very important to state that tables with very few rows are much less affected by the fragmentation percentages.

Therefore, if you use the Index Fragmentation tab and view indexes with large fragmentation percentages, you should also consider the 'Rows in table' and 'Page Count' values. Only tables with high fragmentation percentages AND larger numbers in these columns should be considered for repair.

Some partners and customers have reported that when they have very high numbers in these fields and then repaired (rebuilt) their indexes, they noticed a considerable improvement in the performance of their SYSPRO systems.

## REPAIR ISSUES

There is no automatic repair available in the SQL Health Dashboard relating to Index Fragmentation. This is because index rebuild and reorganization functions should only be performed when the database is not being accessed.

Microsoft suggests the following guidelines:

Fragmentation %	Suggested action
Greater than 30%	Rebuild the index
Between 5% and 30%	Reorganize the index
Less than 5%	Indicates that the statistics are older than 30 days and you should consider updating them

See the following link describing how to perform index Reorganize and Rebuild functions: <https://docs.microsoft.com/en-us/sql/relational-databases/indexes/reorganize-and-rebuild-indexes?view=sql-server-2017>.

See the following link describing how to update your statistics: <https://docs.microsoft.com/en-us/sql/t-sql/statements/update-statistics-transact-sql?view=sql-server-2017>.

See the following link regarding auto creating statistics and the trade-off in performance vs improvements in the query optimizer: <https://docs.microsoft.com/en-us/sql/relational-databases/statistics/statistics?view=sql-server-2017>.

## SQL HEALTH DASHBOARD – SQL USERS

The SQL Users tab shows information about each SQL user and/or login who have access to the selected database.

The list shows Users and/or Logins together with their type and the effective permissions into the database.

Typically, you should expect to see a SYSPRO administrator with Permissions such as **sysadmin** or **dbo** and standard SYSPRO logins with Permissions of **db\_datareader** and **db\_datawriter**.

You should review all SQL logins that have access to each SYSPRO database to ensure that not only each user has the minimum required permissions, but also that additional logins that do not require access are not included.

When used effectively, this helps reduce security access concerns.

There is a topic dedicated to SQL logins and user permissions in this document – see topic: [Configuring SYSPRO to work with SQL Server](#)



However, to briefly summarize here:

- If you are using Windows authentication, then you need to ensure that you have appropriate user permissions to administer each of the system-wide and company specific databases (typically 'sysadmin')
- If you are using SQL Server authentication, then you need a minimum of two SQL users:
  - An administrative user that typically requires 'sysadmin' permissions on all the SYSPRO databases.
  - A standard user that typically requires **db\_datareader** and **db\_datawriter** permissions on all the SYSPRO databases.
  - In addition, you can configure that each SYSPRO operator has their own SQL login – in which case each of these SQL logins also requires **db\_datareader** and **db\_datawriter** permissions on all the SYSPRO databases.

If you notice one or more SQL users with permissions greater than summarized above, then you should investigate whether the SQL Users are required and whether their permissions are greater than required.

## REPAIR ISSUES

There is no automatic repair available in the SQL Health Dashboard relating to SQL Users.

# About the Author

---

This document was authored by Russell Hollick, Chief Software Architect, SYSPRO - Corporate.

Russell has been a member of the SYSPRO development team for over 30 years and is currently responsible for the SYSPRO Architecture team.

This team looks after:

- Database design and the various database access methods
- Client-Server architecture
- Security and authentication models
- eSignatures
- Custom forms
- Report Writer module



Russell was one of the key SYSPRO 8 architects responsible for the SYSPRO 8 scalability and performance improvements, data dictionary and database implementation, including the data conversion from earlier versions of SYSPRO.

Other areas that Russell is passionate about include:

- Agile development
- Code reviews
- Peer learning
- Helping developers and support personnel to get the most out of SYSPRO



[www.syspro.com](http://www.syspro.com)

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

