

SYSPRO Business Activity Queries

SYSPRO 8

Reference Guide

Published: July 2022



CONTENTS

SYSPRO Business Activity Queries

Exploring	1
Starting	4
Solving	5
Using	20

SYSPRO Business Activity Queries

Exploring

Where it fits in?

SYSPRO Business Activity Queries provides a low-code dataset creation tool based on business logic that solves custom dataset requirements across a number of areas of enterprise reporting in SYSPRO.

These include:

- Business insight tiles
- Crystal Sub Reports
- Artificial Intelligence and Machine Learning Dataset Inputs
- Business Intelligence Tools
- SYSPRO Report Writer (via the Data Dictionary)

Data access

Once you save and publish a SYSPRO Business View, it's created within the SQL database and accessible within the business insight tiles. You can also reference it from within the SYSPRO Data Dictionary, which provides access from the SYSPRO Report Writer once the data has been imported into the Report Writer Data Dictionary.

The Data Dictionary is also used by a number of business objects available to third parties (so they can access and retrieve data for customization purposes):

- **COM GENERIC FETCH**¹
- **COM GENERIC FIND**²
- **COM GENERIC BROWSE**³

Because the query interrogates critical business data, security and controlled access are vital to protect the business. Administrators can, therefore, define the business views of SYSPRO data and allow access to appropriate users in a controlled way.



The security access to the SQL views themselves is the same as that for the standard SYSPRO Base Tables. Therefore, any SQL user with `db_datareader` permissions can retrieve data using the views themselves.

¹Business object: COMFCH

²Business object: COMFND

³Business object: COMBRW

Benefits

- Reduced risk of inappropriate access to SYSPRO data.
- Increased flexibility of creating custom linked data, including custom form fields.
- SQL views no longer need to be created outside of SYSPRO (i.e. using a tool like Microsoft SQL Server Management Studio).
- Less reliance on custom SQL objects for data access (for both reporting and integrations).
- Reduced need for SQL knowledge in creating custom datasets.

Navigation

The programs related to this feature are accessed from the **Program List** of the SYSPRO menu:

- *Program List > Administration > General Setup*

Terminology

Business Activity Query (BAQ)

SYSPRO Business Activity Queries are SQL views that are created in the relevant company databases and (because they are defined in SYSPRO) are recognized as data sources. They provide a view into the business data commonly used during the typical business management process. The queries are built and secured in the core ERP system and then surfaced to users via their enterprise reporting and connectivity solutions.

Business insight tiles

Business insight tiles (also called insight tiles or just tiles) are essentially SQL scripts displayed in a tile format within a SYSPRO web view. These tiles are context-driven and let you act instantly on events, facilitating informed business decisions and sound business behavior.

KPIs on tiles can be used to highlight values approaching or exceeding pre-determined objectives or thresholds. This lets you proactively keep an eye on approaching targets. These targets could be a certain value, company-wide goal or a specific target agreed for a person or a role within the organization.

Data source

In the context of SYSPRO Business insight tiles and SYSPRO Business Activity Queries, a data source can be considered as one of the following:

- SYSPRO base table (e.g. [InvMaster](#))
- SYSPRO custom form table (e.g. [InvMaster+](#))
- SYSPRO Business View
- User defined base table (i.e. any base table in the current database that is not part of the standard SYSPRO database)
- User defined custom form table (i.e. any base table defined with a trailing + sign that doesn't form part of the standard custom table definitions)
- User defined views (i.e. any view in the company database that is not defined as a SYSPRO Business View)

SQL View

In SQL, a view is a virtual table whose contents are defined by a query. Like a table, a view consists of a set of named columns and rows of data.

Unless indexed, a view does not exist as a stored set of data values in a database. The rows and columns of data come from tables referenced in the query defining the view and are produced dynamically when the view is referenced.

A view acts as a filter on the underlying tables referenced in the view. The query that defines the view can be from one or more tables or from other views in the current or other databases.

Examples of what views can be used for:

- To focus, simplify and customize the perception each user has of the database.
- As security mechanisms by letting users access data through the view, without granting the users permissions to directly access the underlying base tables of the view.
- To provide a backward compatible interface to emulate a table that used to exist but whose schema has changed.

WYSIWYG (What You See Is What You Get)

WYSIWYG is a system in which editing software allows content to be edited in a form that resembles its appearance when printed or displayed as a finished product.

Starting

Prerequisites

To use this feature, your SQL Admin login must either be a database owner or a member of the DBO role so that they have sufficient permissions to `CREATE`, `ALTER` and `DROP` views in SQL across all your company databases. This ensures that SYSPRO Business Activity Queries are created consistently across all companies so that they can be utilized by entities such as the Report Writer and SYSPRO OData.

Microsoft defines these permissions as follows:

- **Create view** - Requires `CREATE VIEW` permission in the database and `ALTER` permission on the schema in which the view is being created.
- **Alter view** - To execute `ALTER VIEW` (at a minimum) `ALTER` permission on `OBJECT` is required.
- **Drop view** - Requires `CONTROL` permission on the view, `ALTER` permission on the schema containing the view, or membership in the `db_ddladmin` fixed server role.



Solving

FAQs

Creation and maintenance

Can I define a custom column for the SYSPRO Business View?

Yes. Using the {`custom`} option within the **Columns** pane of the **Business Activity Query Builder** program, you can define free-format SQL snippets (up to 3,000 characters) that will be built into the SQL statement for the custom form definition.



Don't include any top-level aggregations or the column alias within the custom column definition snippet. You can rather define these in the **Aggregate** and **Alias** columns of the **Columns** list view.

Can I define a custom WHERE or JOIN clause for the SYSPRO Business View?

Yes. If the standard condition entries available in the Where or Join clause definition of the **Business Activity Query Builder** program don't cover what you require, then you can select the **Custom** function to enter a free-format SQL expression of up to 3,000 characters.

How do I remove a data source from the SYSPRO Business Activity Query?

You can remove a data source from the **Data Sources** list view of the **Business Activity Query Builder** program by highlighting the line and selecting the **Delete** toolbar function (or by pressing **DEL** on your keyboard).

If you delete a data source that has other data sources linked to it, those tables are also removed. As well as any columns associated with these data sources.



An error is displayed if any of the data sources being deleted are referenced in either the Join or Where clauses. In this scenario, you must amend the clauses to remove the references *before* you can remove the data source.

What happens if I change the Alias on a data source?

If you change an alias on a data source in the **Business Activity Query Builder** program (e.g. changing `SorMaster` to `SM`) then the system searches through both the Join and Where clauses and replaces the applicable entries with the new alias. However, if you have a custom join within the SYSPRO Business View, the system only replaces the aliases if they are enclosed in square brackets.

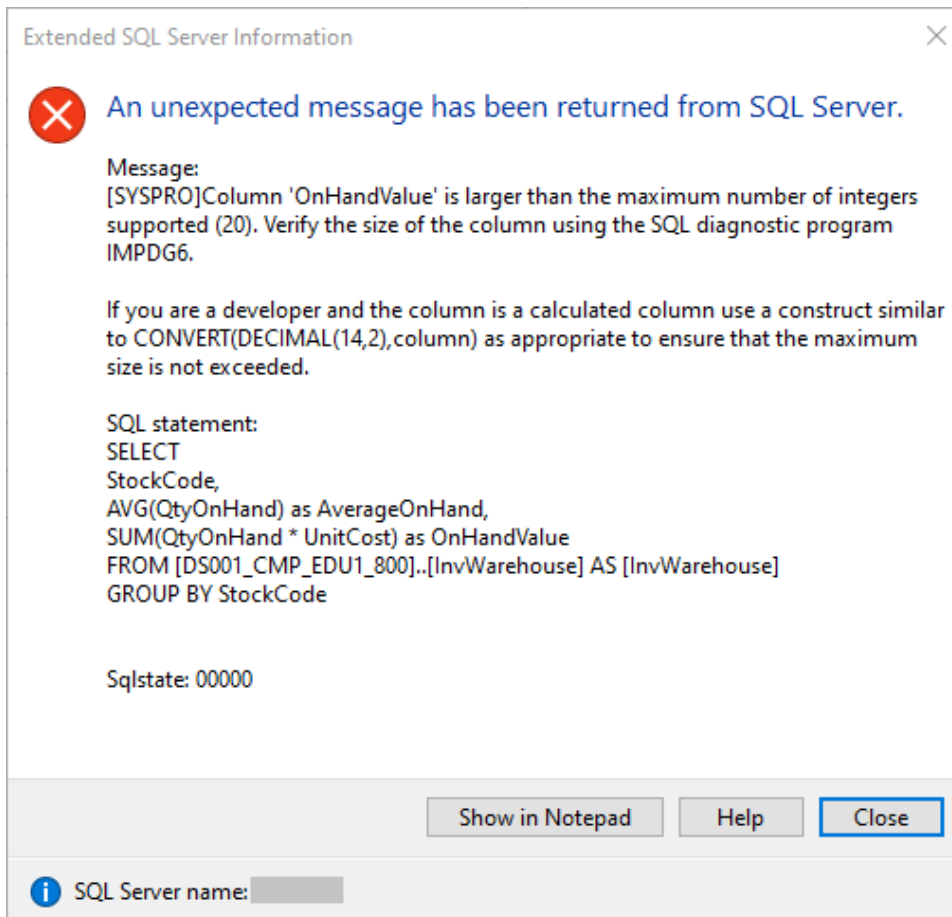
FOR EXAMPLE:

Entries defined as `[SorMaster]` will be replaced with `[SM]`.

What restrictions apply to custom SQL statement column calculations?

The SYSPRO architecture supports a maximum numeric column size of 20 integers and 6 decimals. Therefore, it's quite easy for the implied result to exceed one or both of these limits when a calculation is defined in a SQL statement, as often these are calculated according to the maximum value (i.e. 38 digits).

An error message similar to the following is displayed when you attempt to view the data, if you create a SQL statement with calculated columns that exceeds these limits:



To overcome this, you need to wrap the calculation in either a **CAST** or **CONVERT** function, as per the sample SQL statement below:

```
SELECT
    StockCode,
    CAST(AVG(QtyOnHand) as decimal(20,6)) as AverageOnHand,
    CONVERT(decimal(20,6),SUM(QtyOnHand * UnitCost)) as OnHandValue
FROM InvWarehouse
GROUP BY StockCode
```


Publication

How does the SYSPRO Business View appear in SQL once published?

The SYSPRO Business View's header block within SQL contains the following information:

- Name
- Description
- View version and database version
- Which operator created the view and when
- Which operator last updated the view and when



There are no locking hints defined within the view (e.g. `NOLOCK`) as a view relies on lock inheritance (i.e. the lock at the parent scope is inherited by all resources within that scope). This means that you should define your lock hint when you submit your `SELECT` statement for the view.

The following is a sample definition of a SYSPRO Business View published in SQL (based on the requirement to view the sales order book by customer):

```

/*
SYSPRO Business Activity Query
Name:          bq_SalesOrderBook
Description:   Sales order book by customer
Version:      001( 8.0.0.0028)
Created by:   ADMIN
Date created: 2021/08/19 10:22
Updated by:   ADMIN
Date updated: 2021/08/19 10:22
*/
CREATE VIEW [dbo].[bq_SalesOrderBook] AS
/*BeginSelect*/
SELECT
    [ArCustomer].[Customer] AS [Customer]
  , [ArCustomer].[Name] AS [Name]
  , [SorMaster].[SalesOrder] AS [Sales order]
  , [SorMaster].[OrderDate] AS [Order date]
  , [SorDetail].[MStockCode] AS [Stock code]
  , [SorDetail].[MStockDes] AS [Stock description ]
  , [SorDetail].[MWarehouse] AS [Warehouse]
  , [SorDetail].[MOrderQty] AS [Order Qty]
  , [SorDetail].[MShipQty] AS [Ship Qty]
  , [SorDetail].[MBackOrderQty] AS [Backorder Qty]
  , CASE WHEN [SorDetail].[LineType] = '1' THEN 'Stocked' ELSE 'Non-stocked' END AS [Line type]
FROM [ArCustomer] AS [ArCustomer]
INNER JOIN [SorMaster] AS [SorMaster] ON ([ArCustomer].[Customer] = [SorMaster].[Customer] AND
    [SorMaster].[ActiveFlag] = '' AND
    [SorMaster].[DocumentType] = '0')
INNER JOIN [SorDetail] AS [SorDetail] ON ([SorMaster].[SalesOrder] = [SorDetail].[SalesOrder] AND
    ([SorDetail].[LineType] = '1' OR
    [SorDetail].[LineType] = '7') AND
    ([SorDetail].[MShipQty] <> 0 OR
    [SorDetail].[MBackOrderQty] <> 0))
/*EndSelect*/

```

Auditing

Where can I view an audit of the SYSPRO Business Views published to SQL?

When a SYSPRO Business View is published to SQL (i.e. the view is created or altered in the relevant SQL database) an audit entry is created within the [AdmViewAudit](#) table for each company in which the view has been created.

Therefore, the [AdmViewAudit](#) table holds the following information:

Column	Description	Data Type
Company	Company where view is defined	char(4)
SchemaName	Schema where view is defined	varchar(20)
ViewName	View name	varchar(18)
SaveDateTime	Date and time saved	datetime
VersionView	Version for published view	decimal(6,0)
DatabaseVersion	Database version when published	varchar(20)
Operator	Operator code	varchar(20)
MaintFlag	Function	char(1)

In addition to this auditing capability, the details of the SYSPRO Business View published (i.e. the definition) are held within the following tables:

- [AdmViewMasterPub](#)
- [AdmViewTablesPub](#)
- [AdmViewColumnsPub](#)

Where is SYSPRO Business View information stored in SQL?

The following tables associated with the **SYSPRO Business Activity Queries** feature are retained within the system database and are defined as `MAX` tables (i.e. they don't have associated file handlers and are only accessible via optimized SQL):



The `Company ID` in these tables (with the exception of the `AdmViewAudit` table) is not set, as the SYSPRO Data Dictionary is company agnostic.

Admin Business View Master

[AdmViewMaster](#) is the header (or control) table for SYSPRO Business Views and contains the following information:

Column	Description	DataType
Company	Company where view is defined	char(4)
SchemaName	Schema where view is defined	varchar(20)
ViewName	View name	varchar(18)
Description	Description of the view	varchar(100)
PrimaryDataSource	Primary data source	varchar(19)
ManualEntry	Y - Hand crafted SQL statement	char(1)
Published	Y - View created in SQL	char(1)
Available	Y - View available for use	char(1)
DistinctValues	View is defined to return	char(1)
TopType	Top expression required	char(1)
TopNum	Number in TOP expression	decimal(6,0)
SqlWhere	Where expression	varchar(max)
SqlStatement	Stored SQL statement	varchar(max)
VersionView	Version for published view	decimal(6,0)
FileCode	File code for data dictionary	varchar(6)

Admin Business View Tables

[AdmViewTables](#) contains the data source(s) information that is used in the SYSPRO Business View and contains the following information:



This table only holds a single dummy entry if the view is hand-crafted (i.e. custom).

Column	Description	Data Type
Company	Company where view is defined	char(4)
SchemaName	Schema where view is defined	varchar(20)
ViewName	View name	varchar(18)
TableSequence	Sequence of tables	decimal(3,0)
DataSource	Data Source	varchar(19)
DataSourceAlias	Data source alias	varchar(19)
Description	Data source description	varchar(40)
DatabaseFlag	Company/database flag	char(1)
JoinType	SQL Join type	char(1)
ParentSequence	Parent data source	decimal(3,0)
SqlJoin	SQL join condition	varchar(max)

Admin Business View Columns

[AdmViewColumns](#) contains the column information to be retrieved when using the SYSPRO Business View and contains the following information:



This table is populated by the column names/aliases that are defined in the statement if the view is hand-crafted (i.e. custom).

Column	Description	Data Type
Company	Company where view is defined	char(4)
SchemaName	Schema where view is defined	varchar(20)
ViewName	View name	varchar(18)
TableSequence	Sequence of tables	decimal(3,0)
ColumnSequence	Sequence of columns	decimal(3,0)
ColumnName	Column	varchar(18)
ColumnAlias	Column alias	varchar(18)
Description	Column Description	varchar(35)
AlphaNumericDate	Field type flag	char(1)
ColumnDataType	SYSPRO data type	varchar(18)
SqlDataType	SQL data type e.g. varchar(10)	varchar(16)
AggType	Aggregation types	varchar(10)
OrderSeq	Order by sequence	decimal(2,0)
OrderAscDsc	Order by sequence	char(1)
CustomColumn	SQL Custom column definition	varchar(max)

Admin Published Business View Master

When a SYSPRO Business View is published to SQL, the entries are copied from the relevant definitions tables to [AdmViewMasterPub](#) (i.e. equivalent published table). This is then used to hold the information around the table based on what is actually in SQL.

Therefore, the [AdmViewMasterPub](#) table holds the following information:

Column	Description	Data Type
Company	Company where view is defined	char(4)
SchemaName	Schema where view is defined	varchar(20)
ViewName	View name	varchar(18)
VersionView	Version for published view	decimal(6,0)
Description	Description of the view	varchar(100)
PrimaryDataSource	Primary data source	varchar(19)
ManualEntry	Y - Hand crafted SQL statement	char(1)
Published	Y - View created in SQL	char(1)
Available	Y - View available for use	char(1)
DistinctValues	View is defined to return	char(1)
TopType	Top expression required	char(1)
TopNum	Number in TOP expression	decimal(6,0)
ManualWhere	Y - Manual condition	char(1)
SqlWhere	Where expression	varchar(max)
SqlStatement	Stored SQL statement	varchar(max)
FileCode	File code for data dictionary	varchar(6)

Admin Published Business View Tables

[AdmViewTablesPub](#) contains the details of the data sources used in the published SYSPRO Business View:

Column	Description	Data Type
Company	Company where view is defined	char(4)
SchemaName	Schema where view is defined	varchar(20)
ViewName	View name	varchar(18)
VersionView	Version for published view	decimal(6,0)
TableSequence	Sequence of tables	decimal(3,0)
DataSource	Data source	varchar(19)
DataSourceAlias	Data source alias	varchar(19)
Description	Data source description	varchar(40)
DatabaseFlag	Company / database flag	char(1)
JoinType	SQL Join type	char(1)
ParentSequence	Parent data source	decimal(3,0)
SqlJoin	SQL join condition	varchar(max)

Admin Published Business View Columns

[AdmViewColumnsPub](#) contains the details of the columns available in the published SYSPRO Business View.



This table is used to populate the SYSPRO Data Dictionary for use.

Column	Description	Data Type
Company	Company where view is defined	char(4)
SchemaName	Schema where view is defined	varchar(20)
ViewName	View name	varchar(18)
VersionView	Version for published view	decimal(6,0)
TableSequence	Sequence of tables	decimal(3,0)
ColumnSequence	Sequence of columns	decimal(3,0)
ColumnName	Column	varchar(18)
ColumnAlias	Column alias	varchar(18)
Description	Column Description	varchar(35)
AlphaNumericDate	Field type flag	char(1)
ColumnDataType	SYSPRO data type	varchar(18)
SqlDataType	SQL datatype e.g. varchar(10)	varchar(16)
AggType	Aggregation types	varchar(10)
OrderSeq	Order by sequence	decimal(2,0)
OrderAscDsc	Order by sequence	char(1)
CustomColumn	SQL Custom column definition	varchar(max)

Admin Business View Audit

When a SYSPRO Business View is published to SQL (i.e. the view is created or altered in the relevant SQL database) an audit entry is created within the [AdmViewAudit](#) table for each company in which the view has been created.

Therefore, the [AdmViewAudit](#) table holds the following information:

Column	Description	Data Type
Company	Company where view is defined	char(4)
SchemaName	Schema where view is defined	varchar(20)
ViewName	View name	varchar(18)
SaveDateTime	Date and time saved	datetime
VersionView	Version for published view	decimal(6,0)
DatabaseVersion	Database version when published	varchar(20)
Operator	Operator code	varchar(20)
MaintFlag	Function	char(1)

Business insight tiles

How do I create a new text type business insight tile?

You can either create a new text type business insight tile from scratch, or copy an existing one and then change the tile details.

In this process, we are concentrating on creating a new text type business insight tile:

1. Open the **Insight Tile Definition** program (*SYSPRO Programs > Administration > General Setup*).
2. To create a new tile definition:
 - Select the **New Tile** function.

The **New Tile** window of the **Tile Builder** program is displayed.

3. Configure the following tile details and select **OK**:
 - Tile type - select **Text**



You can only select the tile type if you are creating a new tile, as it is inherited from the copied tile.

- Tile description
 - File name
4. Configure the following sections in the **Tile Builder** program :
 - Tile header and parameters



In the **SYSPRO UI (Avanti)** version the **Configure parameters** link forms part of the **Tile header** section, whereas in the **SYSPRO Desktop** version, the parameters can be configured at the **Parameters** section.

- Summary tile
- Tile preview simulated values

The variables will be replaced at run time using the output of the tile summary SQL script.



This functionality is only available in the **SYSPRO Desktop** version.

5. Select the **Edit Tile Summary** function to configure the SQL script that is used to render the summary tile values – including the main tile value, together with optional subtitle and footer values.

Select the **Edit Tile Detail** function to configure the SQL script that is used to render the column values of the detail list view. These values are displayed when clicking on the tile to zoom into the details.

You can use a simple-to-use interface to have the SQL statement generated, or you can define a custom SQL script, which provides more flexibility when presenting sophisticated information.


Select **Validate SQL and Close** to validate and save the SQL script, and return to the **Insight Tile Definition** program.

6. Select the **Save and Close** function within the **Insight Tile Definition** program.
7. Preview the tile in the **Insight Tile Definition** program.
8. Assign KPIs by selecting the **Add** hyperlink of the applicable column in the **Tiles** list view.



Once the business insight tile has been created, it can be assigned to a workspace using the **Visual Designer** program.

How do I deploy a business insight tile?

1. Select the **Design Web View** (or **Edit Web View**) option from the **Customize and control this application** icon ()



This icon is available on the toolbar of the main SYSPRO window, as well as on each SYSPRO program's toolbar.

2. From the **Visual Designer** program, select the **Add layout section** function.
3. From the **Configure new layout section** window, select **Tile Section**.
4. Optionally enter a **Section title** at the **Widget title** placeholder and assign an icon to the tile if required.
5. Select the **Add section** function.

6. Click in the **Add tile** block (designated by the  icon).

A list of KPI and Metric categories are displayed.



The number of available insights are indicated by a number in brackets alongside each tile category.

7. Drill down into the relevant category and select the tile you want to deploy.
8. (Optional) Define the following for the tile:

- Width
- Text
- Tile background color
- Icon

9. Select the **Add tile** function to assign the tile to the current layout section.

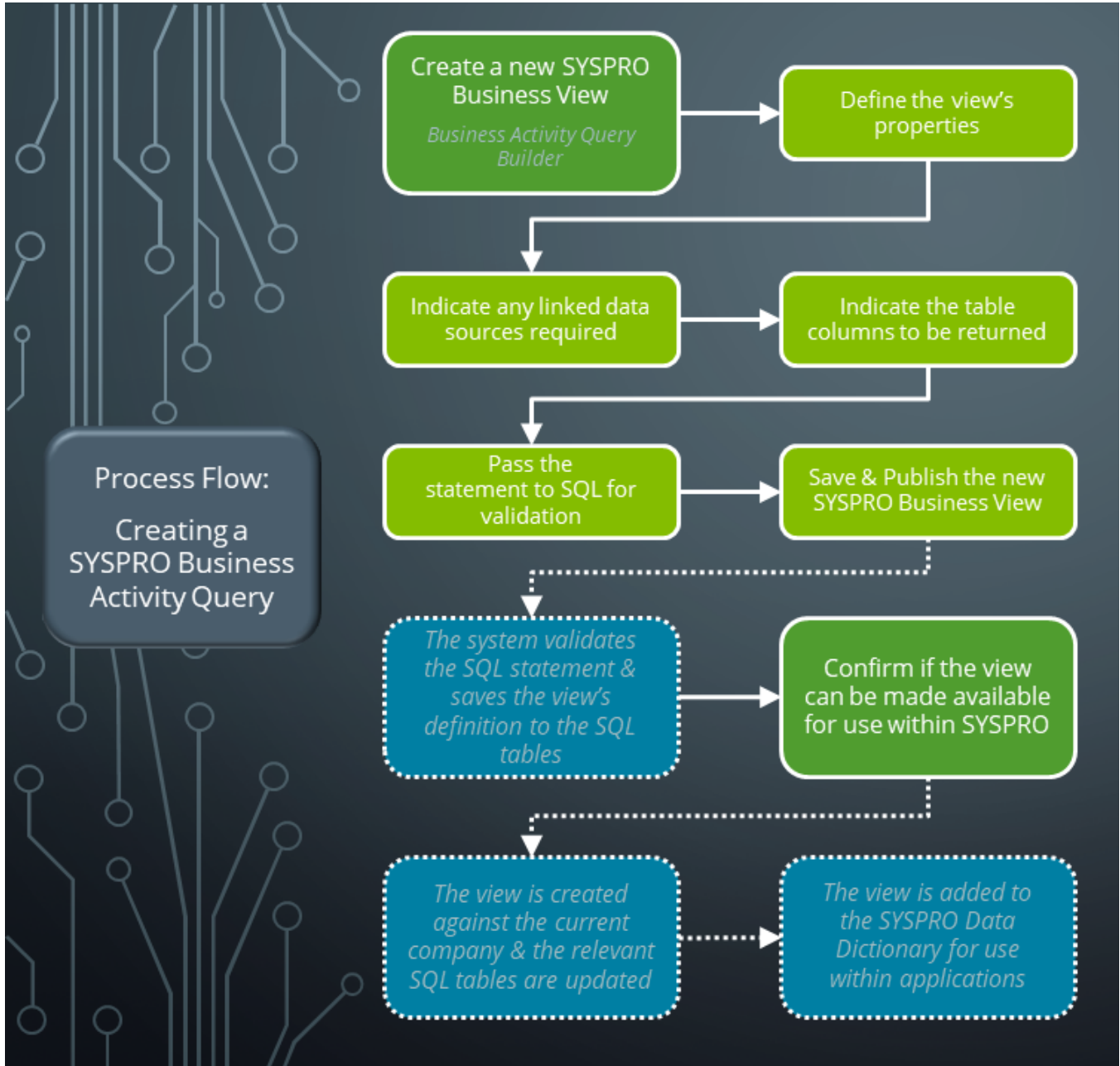
Continue adding tiles to the current layout section or to a new layout section.

10. Select **Save and Exit**.

Your tile will be displayed in the web view, immediately showing the appropriate insight information.

Using

Process



Creating SYSPRO Business Activity Queries

The following describes how to create a new SYSPRO Business Activity Query:

1. Open the **Business Activity Query Builder** program (*Program List > Administration > General Setup*).
2. Within the **View** toolbar field:

Indicate the unique name for the SYSPRO Business View.

FOR EXAMPLE:

bq_ActiveJobs



You can enter up to 18 alphabetic, numeric or underscore characters for the view name but it must be prefixed with **bq_** (i.e. the standard prefix for a SYSPRO Business View). However, the system automatically adds this prefix if you don't include it with your entry.

3. Within the **View Properties** pane, enter the following fields:

Description

Indicate a description for the SYSPRO Business View.

FOR EXAMPLE:

A list of active jobs in SYSPRO

Primary data source

This indicates the 'Anchor' data source around which the SYSPRO Business View is built (i.e. the main data source from which any linked data sources will be defined).

FOR EXAMPLE:

WipMaster

You can search for this by selecting the **Select data source** function. This launches the **Data Sources** program from where you can select the main data source from the **Data Sources** pane.



You can use the **CTRL+F** keyboard shortcut to search for a particular entry.

Once you select the main data source, you are returned to the **Business Activity Query Builder** program.

(Optional) TOP clause and TOP number

This lets you specify the number of rows (or percentage of data) to return on the dataset.



This is useful on large data sets in that it restricts the amount of data returned.

(Optional) Distinct values

This lets you remove any duplicate values from the dataset and return unique rows only.

4. (Optional) Within the **Data Sources** pane, indicate any linked data sources required:



This isn't required if you are only retrieving information from a single source.

Once the **Primary data source** has been entered, it's loaded as the first entry in the **Data Sources** list view. You can then use the data source's hyperlink to indicate additional SQL tables that must be linked to the selected data source.

FOR EXAMPLE:

Primary data source: WipMaster

Linked data source: WipJobClass

After you select a linked data source, the system automatically creates the database table connection to join the information between these sources (i.e. the link between the parent and selected data source is automatically built based on the links defined in the SYSPRO data dictionary).

If the data source selected has no standard link to the parent data source, then you must define the link.

You may also want to refine the link information (e.g. limit job types).



You can optionally change the sequence of the data sources (and columns) by dragging an entry up and down within the grid. However, take care when doing this as the data source sequence is important because SQL works in a top down sequence when validating join information.

5. Within the **Columns** pane, indicate the table columns that you want returned in the SYSPRO Business View:

Data Source

Select the applicable data source (alias) in which the columns that you require reside.



Select `{custom}` if you wish to define a custom column (e.g. complex maths formulas and CASE statements). The **Edit** hyperlink then opens the **Custom Column** window from where you can define the custom SQL statement for the column details.

Columns

Enter the column you want to include from the data source. Alternatively, select the browse function to select multiple database columns at once.

FOR EXAMPLE:

Primary data source: `WipMaster`

Columns: `Job | JobDescription | JobClassification | JobType | StockCode | Customer | Complete | QtyToMake | QtyManufactured`

Repeat this process for each of the linked data sources to ensure that all related column information is included for the SYSPRO Business View.

FOR EXAMPLE:

Linked data source: `WipJobClass`

Columns: `ClassDescription`

6. *Optional...* Define a WHERE clause:



Defining a WHERE clause against the result set is useful as it limits the dataset from returning unnecessary data.

- a. From the **View Properties** pane, select **Edit SQL where clause** within the **WHERE clause** field.

This opens the **Where Clause** pane from where you can define the conditions for the SQL select statement to return rows with distinct column values.

- b. Enter the following fields as required for your clause:

Open

Select the opening brackets required.

FOR EXAMPLE:

- = none

(= one

((= two

Dts1

Select the applicable data source (or `{literal}` entry) from the drop-down.

Column/Literal 1

Enter (or browse for) the column or literal to be compared.



Literals must be enclosed in single quotes if alphanumeric.

Condition

Select the condition required:

- = equal
- <> not equal
- < less than
- > greater than
- <= not greater than
- >= not less than
- Like

Dts2

Select the applicable second data source (or `{literal}` entry) from the drop-down.

Column/Literal 2

Enter (or browse for) the column or literal to compare against.



If you are using a LIKE condition, literals must be set with percentage signs.

FOR EXAMPLE:

'Bob%' = the entry must start with Bob

'%Bob%' = the entry must contain Bob

'%Bob' = the entry must end with Bob

Close

Select the closing brackets required.

FOR EXAMPLE:

- = none

) = one

)) = two

And/Or

If required, select **And** or **Or** to continue the conditions.

FOR EXAMPLE:

To ensure that the results only include uncompleted jobs:

- Open: -
- Dts 1: WipMaster
- Column/Literal 1: Complete
- Condition: <>
- Dts2: {Literal}
- Column/Literal 2: 'Y'
- Close: -
- And/Or: -

7. Select the **Validate SQL** function.

This passes the statement to SQL for validation in the context of creating the view (i.e. it emulates creating a view in SQL).

This is important as it ensures that the statement to be applied doesn't contain invalid statements (e.g. `INSERT`, `UPDATE`, `DELETE`, `TRUNCATE`, etc.).



This validation is performed in the context of the current company and not across all companies where the SYSPRO Business View is to be published. Therefore, if your view contains a custom form column that is defined within the current company, but not within the other companies' databases, then the statement will only be valid in the current company when published (i.e. this can result in a failure during the publishing process to other companies).

8. *Optional...* Select the **Preview** toolbar function.

This lets you preview the data that will be returned before committing any changes to the database or creating the actual SYSPRO Business View.

The **SQL Statement** pane is populated based on your selections once you close the preview.

9. Select the **Save and Publish** function.

This saves the SYSPRO Business View and creates (or alters) the SQL View within the SQL Databases.

During the publishing process, you are prompted to confirm if you want to make the view available for use in SYSPRO. If you select yes, the **Available to use** property of the **View Properties** pane is automatically enabled. The SYSPRO Business View is then immediately accessible within business insight tiles and SYSPRO Data Dictionary.

Publishing SYSPRO Business Activity Queries

The following process occurs when you complete the SYSPRO Business View's definition and select the **Save and Publish** function:



To ensure successful publishing across all required companies, the tables and columns referenced by the SYSPRO Business View must exist in all the company databases that you want to publish the view to.

1. The **Companies to Publish View** window is displayed if the SYSPRO environment has multiple companies, from where you can indicate the companies in which the view must be accessible.

Considerations:

- The SYSPRO Business View is always published to the current company.
- For changes to an existing SYSPRO Business View, the selected entries default to the companies in which the view was previously published.
- If you deselect a company in which an existing view was previously published, then the view is deleted from that company's SQL database.

2. The system validates the SQL statement within the context of the current company to ensure that it doesn't contain invalid statements (e.g. `INSERT`, `UPDATE`, `DELETE`, `TRUNCATE`, etc.).



The publishing process halts and the relevant error message is displayed if this validation fails for any reason.

3. The definition of the SYSPRO Business View is saved to the relevant SQL tables.
4. You are prompted to confirm if you want to make the view available for use in SYSPRO.
If you select yes, the **Available to use** property of the **View Properties** pane is automatically enabled and the SYSPRO Business View is accessible within business insight tiles and SYSPRO Data Dictionary once the publish process has completed.
5. The SYSPRO Business View is created (or altered if it already exists) against the current company.
6. If the publishing process is successful, the version along with the published and available flags are updated within the `Admin Business View Master` table (`AdmViewMaster`).
7. A copy of the view's definition is written to the following SQL tables:
 - Admin Published Business View Master (`AdmViewMasterPub`)
 - Admin Published Business View Tables (`AdmViewTablesPub`)
 - Admin Published Business View Columns (`AdmViewColumnsPub`)

- Admin Business View Audit ([AdmViewAudit](#))

8. When publishing to multiple companies, the SYSPRO Business View is built in the context of each company (considering the collation, shared inventory, shared general ledger, etc.) and is published accordingly.



If the publication fails against a particular company, then the system ignores that company and moves onto the next company for publication.

This can occur if objects (e.g. custom form columns) are not defined in within that particular company.

9. Once all required companies have been processed, the SYSPRO Business View is added (or updated) against the SYSPRO Data Dictionary for use within applications (such as the Report Writer or business objects like **COM GENERIC FETCH**¹ or **COM GENERIC BROWSE**²).

¹Business object: COMFCH

²Business object: COMBRW

Affected programs

The following indicates areas in the product that may be affected by implementing this feature:

Setup programs

Business Activity Query Builder

Program List > Administration > General Setup

This program lets you manage SYSPRO Business Activity Queries (i.e. defined SQL views for information retrieval and custom reporting).

Query programs

SYSPRO Data Viewer

*Accessible from the **View data** function of the **Details** column in the **Data Sources** pane of the **Data Sources** program*

This program lets you view the data contained within a selected data source and is dynamically built based on the data source (e.g. [InvMaster](#)) or SQL SELECT statement it receives from the calling program.

Browse programs

Data Sources

*Accessible from the **Select data source** function of the **Primary data source** field (or when browsing for linked data sources or columns) within the **Business Activity Query Builder** program*

This program lets you view all data sources available within the current SYSPRO company, as well as select an applicable data source or column when creating one of the following:

- Business insight tile
- SYSPRO Business Activity Query



www.syspro.com

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

