opentext™ | File Reporter

Custom Query Guide

Version 26.2

Legal Notices

Condrey Corporation makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Condrey Corporation reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Condrey Corporation makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Condrey Corporation reserves the right to make changes to any and all parts of the software at any time, without obligation to notify any person or entity of such revisions or changes. See the Software EULA for full license and warranty information with regard to the Software.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export, or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. Condrey Corporation assumes no responsibility for your failure to obtain any necessary export approvals.

Condrey Corporation
2131 Woodruff Rd STE 2100 PMB 305
Greenville, SC 29607
U.S.A.

https://condreycorp.com/

Third-Party Systems

The software is designed to run in an environment containing third-party elements meeting certain prerequisites. These may include operating systems, directory services, databases, and other components or technologies. See the accompanying prerequisites list for details.

The software may require a minimum version of these elements to function. Further, these elements may require appropriate configuration and resources such as computing, memory, storage, or bandwidth for the software to be able to perform in a way that meets the customer requirements. The download, installation, performance, upgrade, backup, troubleshooting, and management of these elements is the responsibility of the customer using the third-party vendor's documentation and guidance.

Third-party systems emulating any of these elements must fully adhere to and support the appropriate APIs, standards, and protocols for the software to function. Support of the software in conjunction with such emulating third-party elements is determined on a case-by-case basis and may change at any time.

Contents

About This Guide

The following material provides guidance for developing SQL queries for use with Custom Query reports in File Reporter 26.2. It is intended for network administrators and report developers responsible for creating SQL queries.

1 - Updates and Breaking Changes

1.1 - File Reporter 4.1

1.1.1 - Additional Schema for Microsoft 365

Supported schema for extended Microsoft 365 SharePoint Online data has been added with this release.

A new set of SharePoint-specific tables have been added for improved analysis of permissions in OneDrive for Business and SharePoint Online document libraries.

The new set of tables includes:

- ms365.sp_base_permissions
- ms365.sp_group_members
- ms365.sp_groups
- ms365.sp_permission_levels
- ms365.sp_permissions
- ms365.sp_site_permissions
- ms365.sp_users

In addition, supported references for SharePoint identifiers have been added to the *ms365.permissions* table:

- grantedto_sp_user_id
- grantedto_sp_group_id
- grantedto_sp_login_name
- site_collection_id

1.1.2 - Removed Tables

The *ms365.site_drives* table has been removed as of File Reporter 4.1.

The *ms365.drives* table now include a *site_id* reference column that replaces the need for this bridge table.

Upgrading from File Reporter 4.0 to 4.1 automatically extends this table and populates the corresponding new reference column using the legacy *ms365.site_drives* table before dropping it.

IMPORTANT: Any Custom Queries that reference the legacy *ms365.site_drives* table will need to be updated to make use of the new *ms365.drives.site_id* column instead.
Any queries that continue to reference the legacy table will no longer work after upgrading to File Reporter 4.1 or later until this change has been made.

1.1.3 - Removed Columns

The *grantedto_id_type* string-typed column in the *ms365.permissions* table has been removed as of File Reporter 4.1.

A replacement column *grantedto_type* has been added which is an integer type representing a discrete enumeration.

1.2 - File Reporter 4.0

1.2.1 - Deprecated Views

The following views were deprecated as of File Reporter 4.0 in favor of their corresponding generic view names:

- srs.current_fs_scandata_ad
- srs.previous_fs_scandata_ad
- srs.baseline_fs_scandata_ad

Please use the following views instead, as the *_ad views are subject to removal in a later release:

- srs.current_fs_scandata
- srs.previous_fs_scandata
- srs.baseline_fs_scandata

2 - Supported Constructs

2.1 - Supported Schema Objects

The supported database schema objects include entries in the following categories:

- **Identity Systems:** System name, users, groups, other security principals.
- **Windows File System:** File system metadata and permissions.
- **File Content Analysis Data:** Related to discovery of search expressions over file content.
- **Microsoft 365 Data:** Related to drives, drive items, and supporting metadata and permissions, as well as basic teams and sites info in Microsoft 365.

Although any tables, views, stored procedures, and functions in the database may be accessed via custom queries, only the tables, views, and functions listed here are supported for use with Custom Query development.

NOTE: New SQL users may find the supported views easier to start with as each view provides a simple presentation of several key tables. In addition, the **current_*** views are pre-filtered for only the most recent scan data. Experienced users may find performance benefits in making direct inline queries against the tables themselves, especially for complex queries.

2.2 - Schema Namespaces

All supported database objects and functions reside in specific schema namespaces (e.g., the distinguished name for the table *scan_data*is referenced as *srs.scan_data* when using the namespace prefix).

Always reference each supported database object and function with its documented namespace prefix.

The following table lists the namespaces containing database objects supported for use with custom SQL queries.

Schema Name	Notes
ad	Contains the Active Directory identity data structures.
analysis	Contains file content analysis data structures.

Schema Name	Notes
ms365	Contains Microsoft 365 data structures and functions.
srs	Primary namespace containing all file system data structures and general functions.

2.3 - Supported Tables

Category	Table Name	Notes
Windows File System	srs.identity_systems	List of all identity systems
	srs.ad_objects	List of all scanned Active Directory security principals
	srs.ad_memberships	Active Directory group memberships
	srs.scan_targets	List of all configured scan targets (volumes, shares, etc.)
	srs.scans	List of all available scans
	srs.scan_history	Historical scan summary records
	srs.scan_data	All scan data – includes all path and file-specific metadata info
	srs.scan_directory_data	All directory-specific scan data
	srs.trend_volume_freespace	List of all volume free space records
	srs.ntfs_aces	Scanned NTFS ACEs
	srs.security_descriptors	Scanned NTFS security descriptors

Category	Table Name	Notes
Active Directory	ad.domain	List of scanned Active Directory domains in the forest
	ad.ds_objects	List of scanned security principals in the Active Directory forest
File Content Analysis	analysis.file_scan_entries	Summary classification data for file content analysis entries
Microsoft 365	ms365.drive_items	Files and folders in drives, document libraries
	ms365.drive_item_types	Enumeration table of drive item types
	ms365.drive_scans	List of scans against MS365 drives
	ms365.drive_scans_history	Historical summary of drive scans
	ms365.drives	List of MS365 drives (document libraries, OneDrive for Business drives)
	ms365.group_drives	Mapping of MS365 groups (teams) to associated drives
	ms365.group_member_types	Enumeration table of group member types
	ms365.group_members	MS365 group membership associations
	ms365.group_owners	MS365 group owner associations

Category	Table Name	Notes
	ms365.group_sites	Mapping of MS365 groups (teams) to associated sites
	ms365.groups	List of discovered MS365 groups
	ms365.identity_types	Enumeration table of identity types
	ms365.jobs	List of jobs to enumerate MS365 tenant objects (teams, sites, groups, users, drives, etc.)
	ms365.jobs_history	Historical summary of tenant scans
	ms365.permissions	Sharing links and direct access permissions for drive items
	ms365.sharing_link_members	List of security principals associated with a specific sharing link
	ms365.sites	List of discovered MS365 SharePoint sites
	ms365.sp_base_permissions	Lookup table for SharePoint permission levels / roles.
	ms365.sp_group_members	SharePoint group member associations
	ms365.sp_groups	List of SharePoint groups
	ms365.sp_permission_levels	List of SharePoint permission levels / roles
	ms365.sp_permissions	List of SharePoint permissions (assigned permission levels)
	ms365.sp_site_permissions	List of SharePoint site permissions
	ms365.sp_users	List of SharePoint users

Category	Table Name	Notes
	ms365.team_channels	List of discovered Teams Channels
	ms365.teams	List of discovered MS365 Teams
	ms365.tenants	Configured MS365 tenants for scan
	ms365.user_drives	Mapping of MS365 users to drives (OneDrive for Business drives)
	ms365.users	List of discovered MS365 users
Session-Specific	tmp_cq_fs_paths	Temporary table injected into custom query sessions for report-defined target paths

2.4 - Supported Views

Category	View Name	Notes
Windows File System	srs.current_fs_scans	List of Current file system scans
	srs.current_permissions_scans	List of Current permissions scans
	srs.previous_fs_scans	List of Previous file system scans
	srs.previous_permissions_scans	List of Preview permissions scans
	srs.baseline_fs_scans	List of Baseline file system scans
	srs.baseline_	List of Baseline permissions scans

Category	View Name	Notes
	permissions_scans	
	srs.current_fs_scandata	List of all Current file system scan data
	srs.previous_fs_scandata	List of all Previous file system scan data
	srs.baseline_fs_scandata	List of all Baseline file system scan data
	srs.current_ntfs_aces	All Current permissions scan data for NTFS-compatible file systems
	srs.previous_ntfs_aces	All Previous permissions scan data for NTFS-compatible file systems
	srs.baseline_ntfs_aces	All Baseline permissions scan data for NTFS-compatible file systems
Active Directory	ad.ds_objects_view	All primary properties from ad.ds_objects and ad.domains with binary GUIDs and SIDs converted to equivalent text variants.

2.5 - Supported Functions

Category	View Name	Description
General	srs.byte_string	Converts raw number to byte string such as 10 MB or 3.25 KB
	srs.byte_unit_string	Converts raw number to byte string with specified unit and precision.
	srs.attribute_string	Converts attributes to string representation

Category	View Name	Description
	srs.guid_bytes	Converts GUID from string to binary
	srs.guid_text	Converts GUID from binary to string
	srs.path_hash	Calculates SHA-1 hash of lowercase input (typically a path)
	srs.path_hash_sha256	Calculates SHA256 hash of lowercase input (typically a path)
	srs.bytes_to_hex_string	Converts byte array to equivalent hex string
	srs.hex_string_to_bytes	Converts hex string to equivalent byte array
Identity Systems	srs.sid_bytes	Converts SID from string to binary
	srs.sid_text	Converts SID from binary to string
	srs.ad_account_name	Combines AD account name elements into a single display name
Permissions	srs.access_mask_basic_string	Converts access mask value to basic permissions string
	srs.access_mask_string	Converts access mask value to string representation
	srs.ace_flags_string	Translates ACE flag to string values
	srs.ace_type_string	Translates ACE type to string value

3 - Navigating Scan Data

Writing useful and accurate queries requires a proper understanding of how to navigate collected scan data.

Due to how File Reporter collects and curates scan data, this section is broken up by resource type. It also provides guidance on how to report across these resource types in a single report query, when applicable.

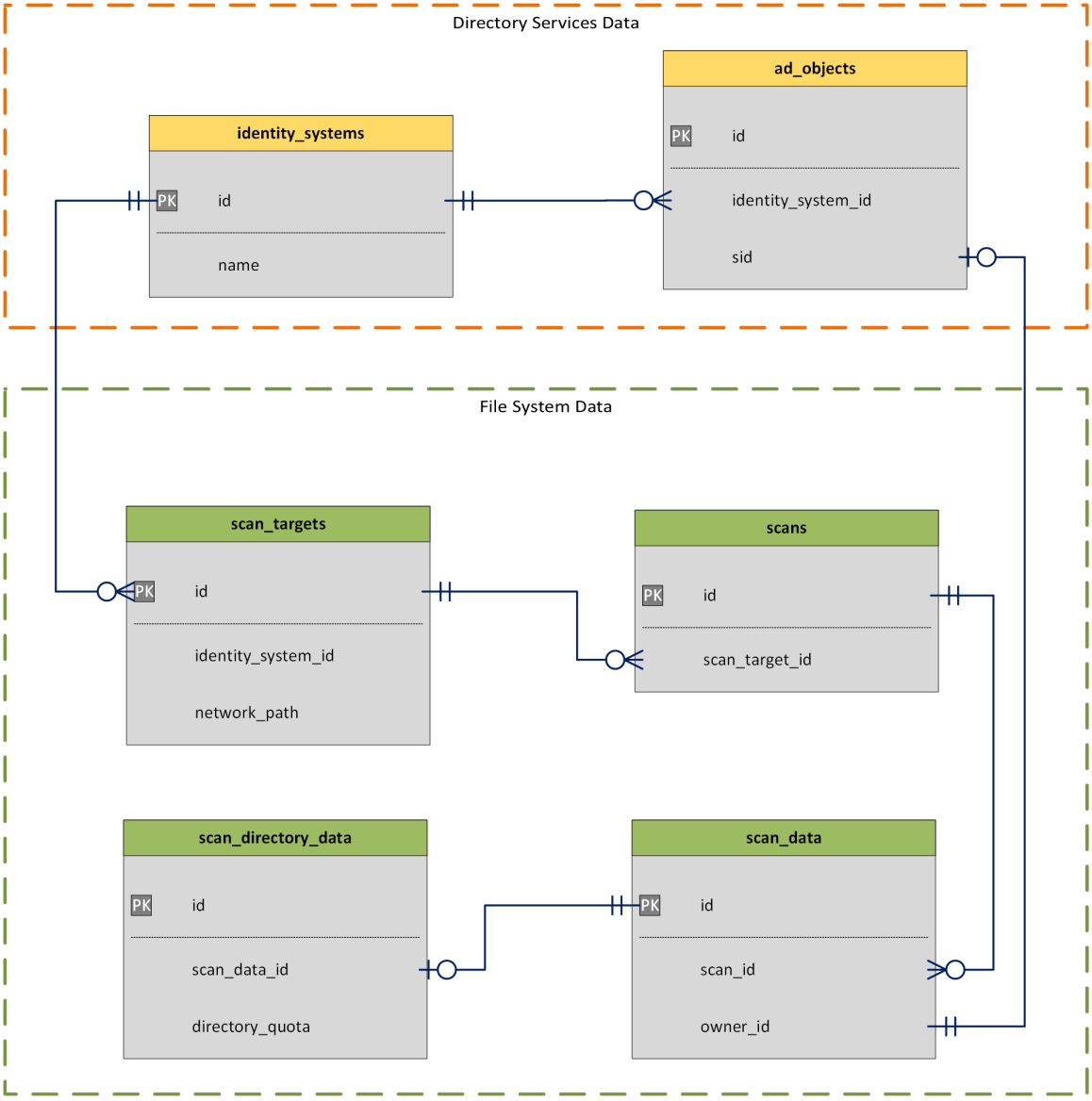
3.1 - Windows File System

3.1.1 - Table Relationships

Windows File System Metadata

Collected scan data is broken down into three major categories: Identity System Data, File System Data, and Permissions Data.

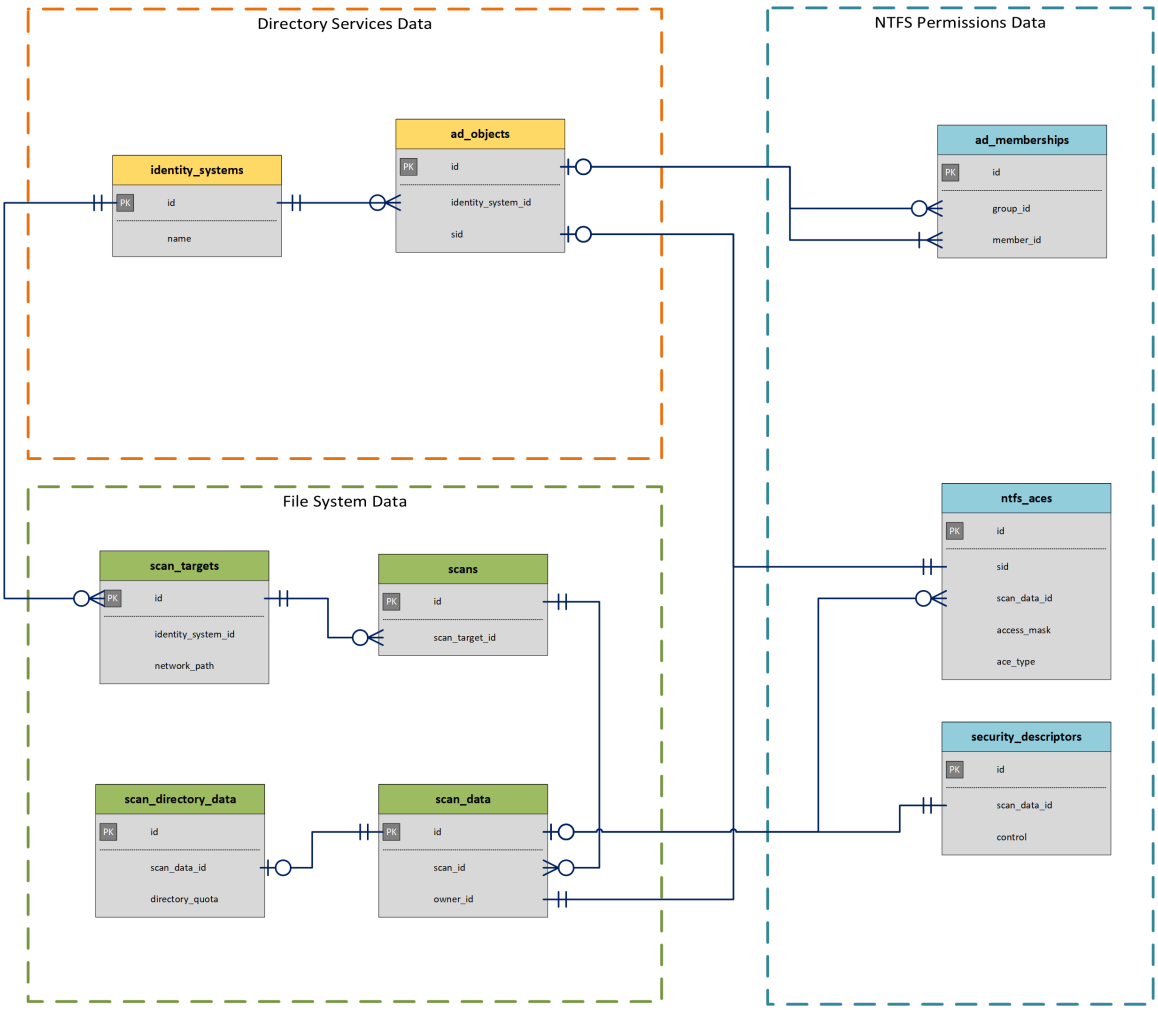
For general file system metadata collection, File System Data is collected along with minimal identity system data pertaining to file and folder owners.



Windows File System Permissions

NTFS Permissions Data is limited to folder structures, and assigned and inherited NTFS access control entries (ACEs).

Permissions scans do not include metadata-specific information such as directory quota, nor do they include any file-entry data that is not a folder. Only permissions for folder, share, and DFS entries are collected currently.



3.1.2 - Scoping and Filtering

Scoping is the process by which selected data is limited to areas of interest. Areas of interest may include all file system data related to a specific identity system, or only data within one or more subdirectories. Additionally, data could be scoped as it relates to a given owner or trustee.

Scope by Identity System

Scoping by identity system is as simple as limiting a query to a specific *srs.identity_system.id* value, or using one of the supported *srs.current_** views, a specific identity system name.

The following example selects file system data from a given identity system, limited to 100 entries.

Example (SQL Server)

```
SELECT TOP(100) *
FROM srs.current_fs_scandata
WHERE identity_system = 'ad.test.lab';
```

Example (PostgreSQL)

```
SELECT *
FROM srs.current_fs_scandata
WHERE identity_system = 'ad.test.lab'
LIMIT 100;
```

Scope by Server

Scoping by server is as simple as filtering by the server column in the *srs.scan_targets* table or in one of the supported *srs.current_** views.

Also note that the server name may be case sensitive depending on the database collation.

The following example selects all file system data from a specific server, limited to 100 entries.

Example (SQL Server)

```
SELECT TOP(100) *
FROM srs.current_fs_scandata
WHERE server = 'server1.ad.test.lab';
```

Example (PostgreSQL)

```
SELECT *
FROM srs.current_fs_scandata
WHERE server = 'server1.ad.test.lab'
LIMIT 100;
```

Scope by Scan Target

Scoping by scan target is useful where a specific CIFS share name or DFS target is known.

Note that the scan target name may be case sensitive depending on the database collation.

Example: select file system data from a particular scan target (share or volume) limited to 100 entries

Example (SQL Server)

```
SELECT TOP(100)
  *
FROM srs.current_fs_scandata
WHERE scan_target = '\\server1.ad.test.lab\Data';
```

Example (PostgreSQL)

```
SELECT
  *
FROM srs.current_fs_scandata
WHERE scan_target = '\\server1.ad.test.lab\Data'
LIMIT 100;
```

Scope by Directory

Scoping by a particular directory or folder requires the use of the hierarchical markers in the *srs.scan_data* table.

These markers assist with determining parent and child folders as well as all subordinate file system entries for a given directory or set of directories.

Field	Description	Notes
idx	Entry index	Unique per scan
parent_idx	Index of parent directory, share, or DFS name space entry	All sibling file system entries will have the same parent index.
path_depth	Current path depth relative to the root path	The root path is always depth zero (0). Other paths such as shares may have the same depth as the root path, but can be distinguished by *path_type*. Entries occurring above the root path (such as

Field	Description	Notes
		DFS name spaces) will have a negative value.
ns_left **ns_right**	Nested set indexes for current entry	Nested set markers provide a quick way to determine all subordinates for a given directory. See examples below for details.

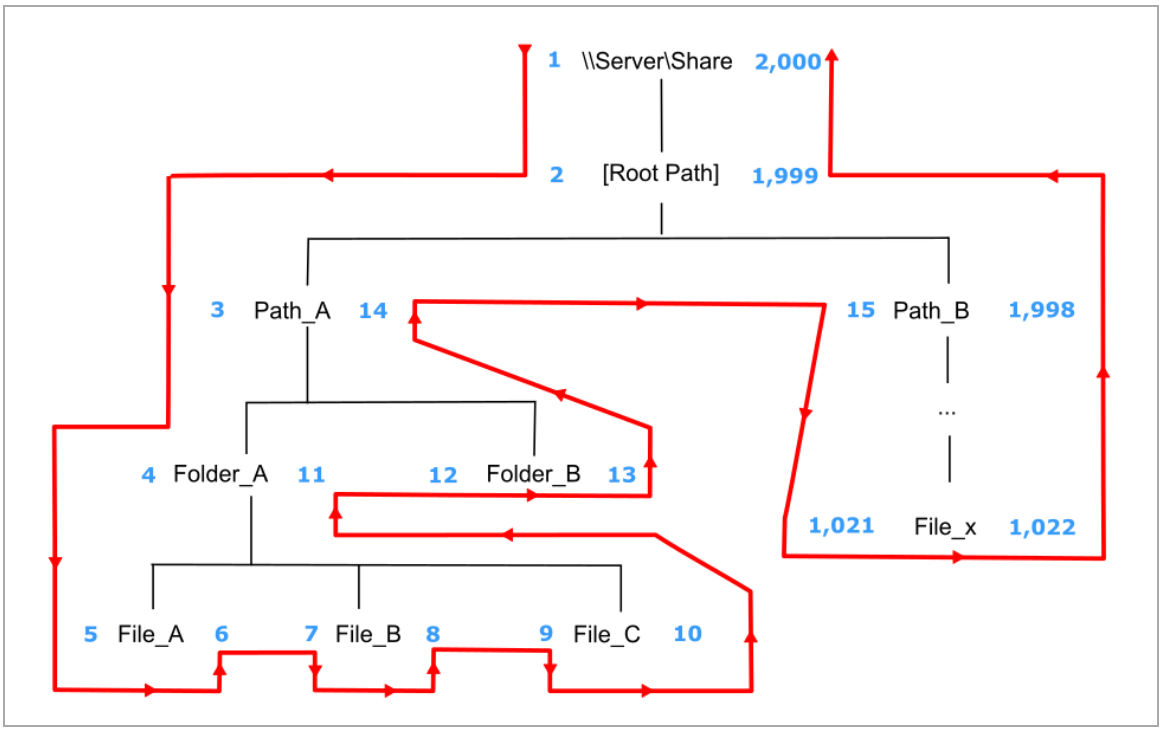
The following example selects all NTFS file system entries subordinate to and including the specified target path.

Example - Scope by Directory

```
1  WITH root_path AS (
2    SELECT
3      sd.ns_left,
4      sd.ns_right,
5      sd.scan_id
6    FROM srs.current_fs_scandata_ad AS sd
7    WHERE sd.fullpath_hash = srs.path_hash
   ('\\server1.ad.test.lab\Share\path\subpath')
8      AND sd.path_type = 2
9  )
10 SELECT
11   sd.*
12 FROM srs.current_fs_scandata_ad AS sd
13 JOIN root_path AS rp ON rp.scan_id = sd.scan_id
14  AND rp.ns_left <= sd.ns_left
15  AND rp.ns_right >= sd.ns_right;
16
```

In this example, we are using two SELECT statements: one to get the information for the desired root path, and one to pull all subordinate entries along with the root path. Notice how the JOIN filter in the second SELECT statement uses not only the *scan_id* to limit the particular scan(s) of interest, but also uses the *ns_left* and *ns_right* fields to keep the data set limited to file entries in the folder hierarchy.

In the following diagram, an example of the nested set model calculations are shown with an example structure under \\Server\Share. In this example, exactly 1,000 file system entries exist, including files, folders, and the share itself.



For each node in the scanned file structure, a left (*ns_left*) and right (*ns_right*) value are assigned. The values are assigned by traversing the imaginary path from the root down the left side of the structure, incrementing the *ns_left* values by one. Once a leaf node is encountered, the incrementing value continues, but is now assigned to *ns_right*.

This process continues until the entire graph of the file structure has been traversed, and the root path is finally assigned the last number for its *ns_right* value.

The nested set model has the following characteristics, some of which are vital to hierarchical processing, such as determining subordinate objects:

• The root path will always have a *ns_left* value of 1 and an *ns_right* value of 2n, where n = the total number of entries

• For any given container object (folder, share, etc.), all subordinate entries can be found by searching for all objects in the scan having an *ns_left* value greater than the container path's *ns_left* value, and an *ns_right* value less than the container path's *ns_right* value.

• Nested set is generally the fastest method available in relational data models for retrieving all subordinate objects when representing hierarchical data.

For more information on the nested set model, see https://en.wikipedia.org/wiki/Nested_set_model.

Scope by Directory with Path Depth

In addition to scoping by directory, it may be useful to start with a given path, but then only include subordinate paths within a given range below the selected path.

In this case, we make use of the same nested set model calculations seen in the previous section, but include the use of the path_depth parameter as well.

The following example selects all paths starting two levels below a given path:

Example - Start with Path Depth 2

```
WITH root_path AS (
  SELECT
    sd.ns_left,
    sd.ns_right,
    sd.scan_id,
    sd.path_depth
  FROM srs.current_fs_scandata_ad AS sd
  WHERE sd.fullpath_hash = srs.path_hash('\\server1.ad.test.lab\Share\Groups')
  AND sd.path_type = 2
)
SELECT sd.*
FROM srs.current_fs_scandata_ad AS sd
JOIN root_path AS rp ON rp.scan_id = sd.scan_id
 AND rp.ns_left <= sd.ns_left
 AND rp.ns_right >= sd.ns_right
 AND sd.path_depth > rp.path_depth + 2;   -- Upper bound
```

This example is common when folder structures have managed content, such as collaborative or group folders, organized below division or department folders one or more layers deep.

In order to pull all the content from just the group folders themselves, and not include the structural folders, we can make use of path depth, but assign the selected path to the root structural folder.

For a share organized as:

\\Server\Share\Groups\Departments\GroupA

the selected path could be \\Server\Share\Groups and the *path_depth* could be assigned to the *root_path + 2* or greater, as in the SELECT statement above.

We could just as easily limit the depth of paths searched by adding another comparison of *path_depth* as a lower bounds:

Example - Upper and Lower Path Depth

```
WITH root_path AS (
  SELECT
```

```
    sd.ns_left,
    sd.ns_right,
    sd.scan_id,
    sd.path_depth
  FROM srs.current_fs_scandata_ad AS sd
  WHERE sd.fullpath_hash = srs.path_hash('\\dbdev.db.dtest.lab\home')
  AND sd.path_type = 2
)
SELECT sd.*
FROM srs.current_fs_scandata_ad AS sd
JOIN root_path AS rp ON rp.scan_id = sd.scan_id
 AND rp.ns_left <= sd.ns_left
 AND rp.ns_right >= sd.ns_right
 AND sd.path_depth > rp.path_depth + 2   -- Upper bound
 AND sd.path_depth < rp.path_depth + 3;  -- Note that we have a lower bound as well
```

Scope by Security Principal

Scoping by security principal is useful when querying for scan data specific to a given set of owners or trustees.

This example selects all files for a given server owned by a specific AD user, limited to 100 entries.

Example (SQL Server)

```
SELECT TOP(100) *
FROM srs.current_fs_scandata_ad
WHERE owner_domain = 'AD'
  AND owner_name = 'user1';
```

Example (PostgreSQL)

```
SELECT *
FROM srs.current_fs_scandata_ad
WHERE owner_domain = 'DB'
  AND owner_name = 'test1'
LIMIT 100;
```

This next example selects all folders where a user is a direct trustee (not inherited) for NTFS folders, limited to 100 entries.

Example (SQL Server)

```
SELECT TOP(100) *
FROM srs.current_ntfs_aces
WHERE trustee_domain = 'DB'
  AND trustee_name = 'test1'
  AND ace_flags & 16 <> 16;
```

Example (PostgreSQL)

```
SELECT *
FROM srs.current_ntfs_aces
WHERE trustee_domain = 'DB'
  AND trustee_name = 'test1'
  AND ace_flags & 16 <> 16
LIMIT 100;
```

Basic Filtering

In addition to using filters to scope the range of scan data, basic filtering can also be used to limit the results to only records of interest.

The following is a list of basic filtering examples that may be used as starting templates for queries.

Filter by Path Type

In cases where aggregation or calculations against a discrete set of files is desired, it may be necessary to filter out any directories or shares first, since those entries contain size and name data that may skew the desired results.

```
SELECT *
FROM srs.current_fs_scandata_ad
WHERE path_type = 1          -- Note: 1 = file entry
  AND server='Server1';
```

Filter by File Extension

This example filters the set of file entries within a given directory structure to just those defined as media types.

```
SELECT *
FROM srs.current_fs_scandata_ad
```

```
WHERE path_type = 1
  AND filename_extension IN ('mp3', 'mp4', 'avi', 'ogg', 'png', 'jpg', 'jpeg');
```

Note that for *filename_extension*, all values should be lower case.

Filter by Date Range

This example selects all files on the specific server from November 1, 2013 midnight, through November 2, 2013 11:59 PM.

```
SELECT *
FROM srs.current_fs_scandata_ad
WHERE modify_time BETWEEN '2013-11-01 00:00:00' AND '2013-11-02 23:59:59'
  AND server='dbdev.db.dtest.lab'
  AND path_type = 1   -- Files only
```

We can also use the familiar >= and <= comparison operators to accomplish the same:

```
SELECT *
FROM srs.current_fs_scandata_ad
WHERE modify_time >= '2013-11-01 00:00:00'
  AND modify_time <= '2013-11-02 23:59:59'
  AND server='dbdev.db.dtest.lab'
  AND path_type = 1   -- Files only
```

Note that the behavior of the BETWEEN operator is inclusive, not exclusive, to the parameters given.

It is important to note with date-time ranges, that a simple date such as ‘2013-11-02’ actually represents ‘2013-11-02 00:00:00’, so be careful to include 23:59:59 to the ending date as appropriate.

Finally, it is important to remember that all timestamps stored in the database are stored as UTC values, so consideration for time zone offsets may be needed.

Filter by File Name

This example shows how to filter by a given file name.

```
SELECT *
FROM srs.current_fs_scandata
WHERE LOWER(name) = 'document1.txt';
```

Note the use of the LOWER operator to force a case-insensitive search. Depending on the collation of the database instance and the database itself, this operator may be required.

For wildcard matches, the standard SQL flags _ and % can be used to represent single or multiple characters.

```
SELECT *
FROM srs.current_fs_scandata
WHERE LOWER(name) LIKE 'document1.%';
```

See the following links for database specific info regarding wildcards and other search patterns:

- SQL Server: https://msdn.microsoft.com/en-us/library/ms190301
- PostgreSQL: https://www.postgresql.org/docs/current/static/functions-matching.html

3.1.3 - File System Target Paths

You can define and manage a Custom Query report's selected target paths via the report definition itself, separate from any associated SQL queries.

This is accomplished via a temporary table that is injected into the SQL query session at runtime when using any of the File Reporter tools such as Report Designer or the SQL query editor in the File Reporter Web Application for Custom Query reports.

Newer report templates available on the File Query Cookbook site (https://filequerycookbook.com) make use of this feature, which provides a more hands-off approach to modifying SQL queries directly, but with the flexibility to define and change a report's file system target paths.

Example Query

The following example illustrates a custom query that reports on NTFS file system permissions for one or more target paths selected with the *File System Target Paths* dialog in Report Designer.

IMPORTANT: SQL Server requires a hash '#' prefix to reference temporary tables. When using SQL Server as the backend database, be sure that any references to **tmp_cq_fs_paths** in your SQL queries are changed to **#tmp_cq_fs_paths** instead.

Conversely, PostgreSQL can't use hash marks '#' as part of the table name, so be sure that this prefix does not exist in your SQL queries when using PostgreSQL as the backend database.

1. Launch the File Reporter Report Designer application and create a new empty report – see Creating a Report in the *File Reporter Client Tools Guide* for details.
2. Enter one of the following SQL queries into the SQL query editor dialog, depending on the database in use.

Example (PostgreSQL)

```
SELECT
 *
FROM srs.current_ntfs_aces AS ace
JOIN tmp_cq_fs_paths AS cq
 ON cq.target_path_hash = ace.fullpath_hash
 AND cq.is_current = 'true'
 AND cq.is_permission_scan = 'true';
```

Example (SQL Server)

```
SELECT
                *
                FROM srs.current_ntfs_aces AS ace
                JOIN #tmp_cq_fs_paths AS cq
                ON cq.target_path_hash = ace.fullpath_hash
                AND cq.is_current = 'true'
              AND cq.is_permission_scan = 'true';
```

3. Click *Save* to save the SQL query.
4. Click *File System Paths* to open the File System Target Paths dialog.
5. Select one or more paths to report on, then save the selection. Be sure to select paths marked as having Permissions scan data available, as seen in the *File System Target Paths* dialog.
6. Click *Execute Query* to run the SQL query and view the results.

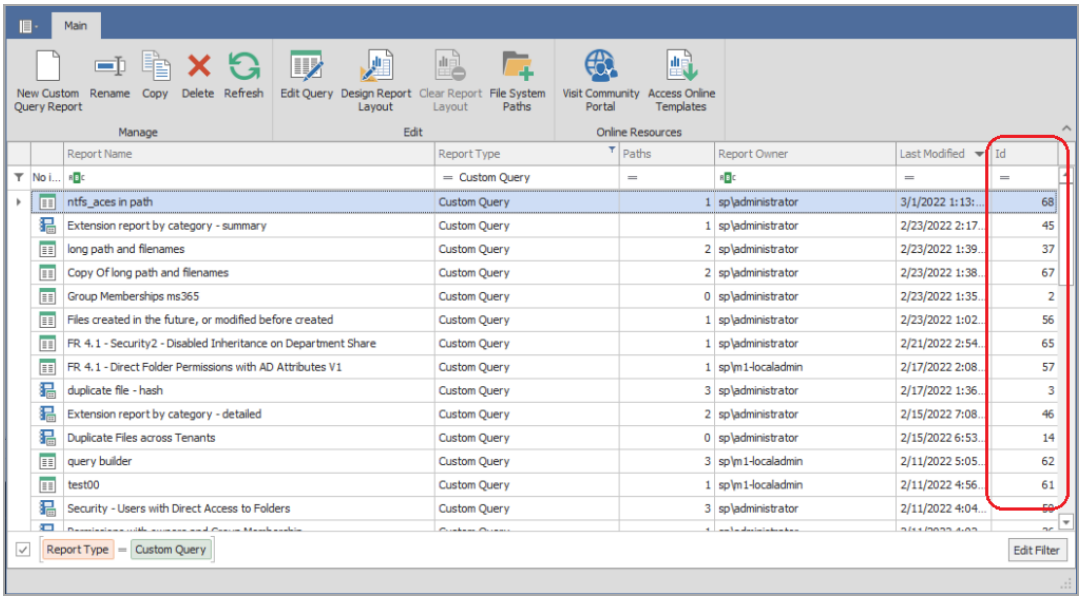
Using Alternate SQL Query Editors

You can choose to develop a SQL query for a Custom Query report in a query editor of your choice, such as PgAdmin for PostgreSQL or SQL Server Management Studio (SSMS).

In these development environments, however, the injected temporary table is not available by default. Use the following procedures to stage the temporary table.

IMPORTANT: Although any existing report definition may be used as a reference, you should create a new Report Definition and use its associated ID. Doing so gives you the flexibility to change the selected target paths during the query design phase without impacting other report definitions.

1. Create a new Custom Query report – see Creating a Report in the *File Reporter 26.2 Client Tools Guide* for details.
2. Assign one or more File System target paths to the report definition – see File System Paths Selector in the *File Reporter 26.2 Client Tools Guide* for details.
3. Find the report ID for the newly-created report.



   a. Find the name of the newly-created report definition in the Main window of the Report Designer.
   b. The column at the far right of the grid indicates each report's ID. Make note of the new report definition's ID number.

4. Insert the following SQL code at the start of the query:

Example (PostgreSQL)

```
CREATE TEMP TABLE IF NOT EXISTS tmp_cq_fs_paths AS
SELECT * FROM srs.cq_fs_paths_by_report_id(17);
```

Example (SQL Server)

```
IF OBJECT_ID('#tmp_cq_fs_paths', 'U') IS NULL
                SELECT * INTO #tmp_cq_fs_paths
```

```
3 |             FROM srs.cq_fs_paths_by_report_id(17);
```

5. Change the example's report ID of "**17**" to the report ID identified from the previous step.
6. Add SQL statements as needed to complete the query.
7. When the SQL query development is complete, copy all the SQL statements into the Custom Query report definition, except for the initial lines used to stage the temporary table .

Using the earlier example query, a complete query using a staged temporary table with an alternate SQL query editor looks like this:

Example (PostgreSQL)

```
1  | CREATE TEMP TABLE IF NOT EXISTS tmp_cq_fs_paths AS
2  | SELECT * FROM srs.cq_fs_paths_by_report_id(17);
3  |
4  | SELECT
5  |  *
6  | FROM srs.current_ntfs_aces AS ace
7  | JOIN tmp_cq_fs_paths AS cq
8  |  ON cq.target_path_hash = ace.fullpath_hash
9  |  AND cq.is_current = 'true'
10 |  AND cq.is_permission_scan = 'true';
```

Example (SQL Server)

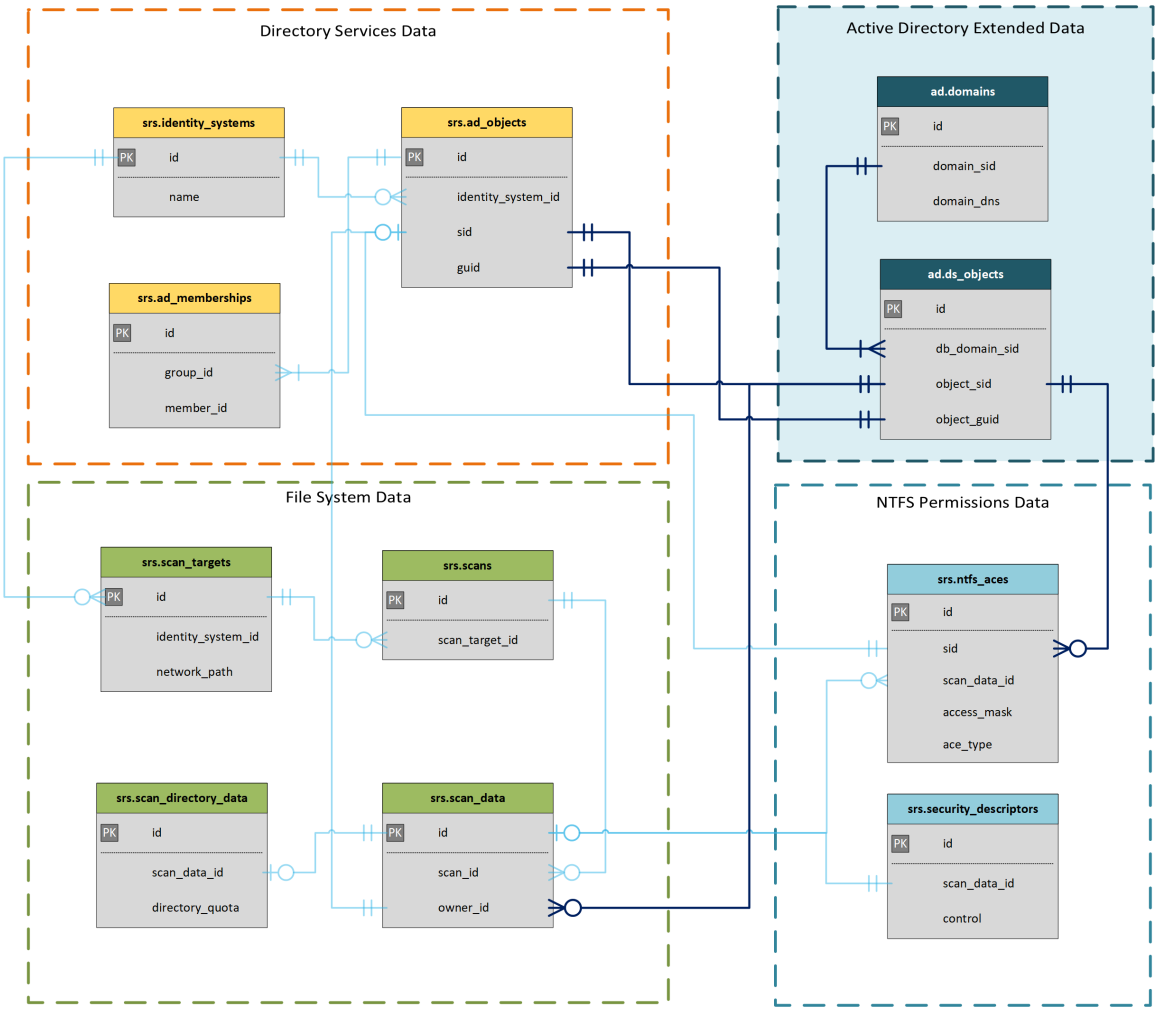
```
1  | IF OBJECT_ID('#tmp_cq_fs_paths', 'U') IS NULL
2  |         SELECT * INTO #tmp_cq_fs_paths
3  |         FROM srs.cq_fs_paths_by_report_id(17);
4  |
5  |         SELECT
6  |         *
7  |         FROM srs.current_ntfs_aces AS ace
8  |         JOIN #tmp_cq_fs_paths AS cq
9  |         ON cq.target_path_hash = ace.fullpath_hash
10 |         AND cq.is_current = 'true'
11 |       AND cq.is_permission_scan = 'true';
```

3.2 - Active Directory Identities

The extended data for Active Directory identities is stored in the *ad.domains* and *ad.ds_objects* tables.

The tables used to map basic identity information for owners and permission trustees may be joined to these tables to extend the data further.

Note that while the current extended Active Directory information does not yet include group memberships, you may continue to use the existing group membership table *srs.ad_memberships* that identifies group members for discovered file system trustees.



See the example scenario *Active Directory Identity Enrichment (page 37)* in this guide for an example of integrating the extended Active Directory identity data in a Custom Query.

4 - Example Scenarios

4.1 - Content Hash Duplicate File Reports

A Content Hash Duplicate File report provides more advanced duplicate file detection over the Duplicate File built-in report which compares only file names and metadata.

NOTE: For information on collecting content hashes, see Creating A Scan Policy in the *File Reporter 26.2 Administration Guide*.

Through https://filequerycookbook.com you can copy and paste the Content Hash Duplicate File Report custom query into the Query Editor and import a report layout into the Report Designer. This custom query and associated report identifies duplicate files based on hash comparisons and the parameters you set.

4.1.1 - Determining Prerequisites

- Create a file system scan policy for each of the target paths on which you want to report.
- With the *Generate content file hashes* option selected in the Scan Policy Editor of each scan policy, conduct a file system scan on each target path.
- Install the Client Tools.

  The Client Tools include the Query Editor and the Report Designer that will be used in these procedures.
- Decide how you want the report to be generated and follow the applicable procedures.
  - To generate a delimited text file that you can take into other tools for customized searching and presentations you can copy or create an SQL query with the query editor covered in Creating a Report in the *File Reporter 26.2 Client Tools Guide*
  - To generate the report using the Report Designer and produce a formatted report layout, proceed with Using Report Designer in the *File Reporter 26.2 Client Tools Guide.*

4.1.2 - Designing the Report

This option lets you utilize both the custom query and the associated report layout design for the “Content Hash Duplicate File Report” from https://filequerycookbook.com.

NOTE: A detailed discussion of the Report Designer along with procedures for familiarizing yourself with the interface are available in Using Report Designer in the *File Reporter 26.2 Client Tools Guide.*

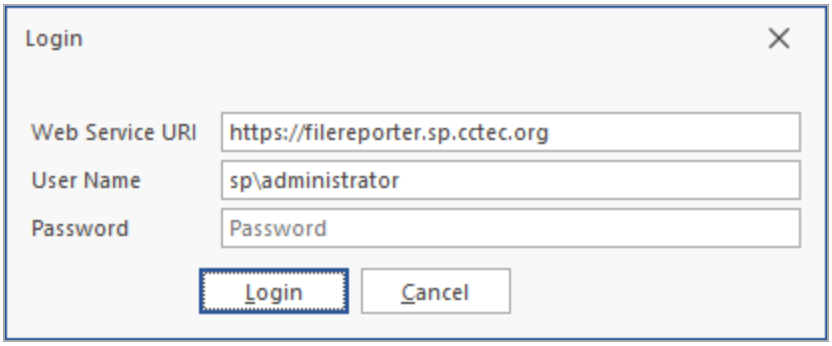
1. On the File Query Cookbook site https://filequerycookbook.com locate and download the “Content Hash Duplicate File Report.”

   The file is saved as a zip archive.

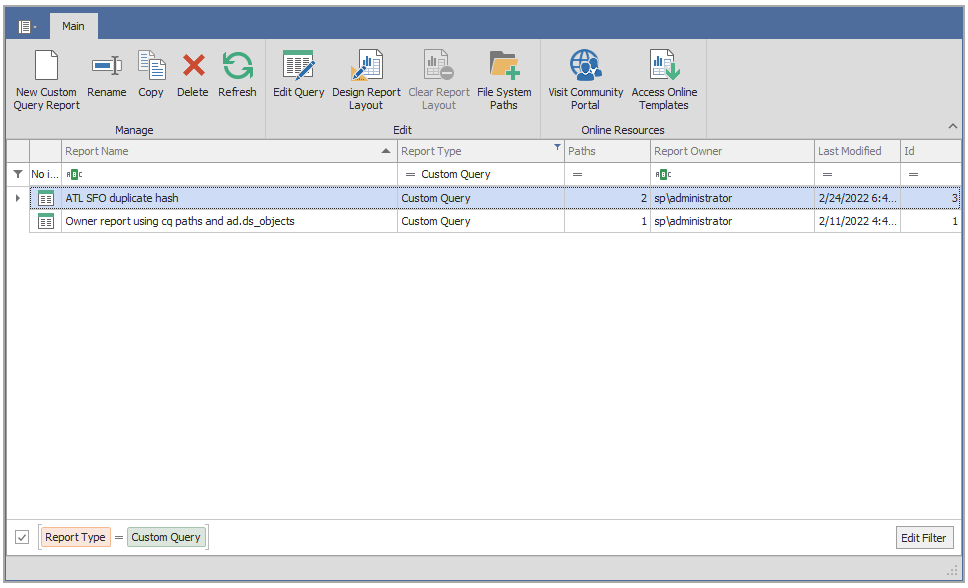
2. Unzip the downloaded file and open the `.sql` file in a text editor.

   You will eventually paste this custom query into the Query Editor.

3. From the *Start* menu, launch the File Reporter 26.2 Report Designer.

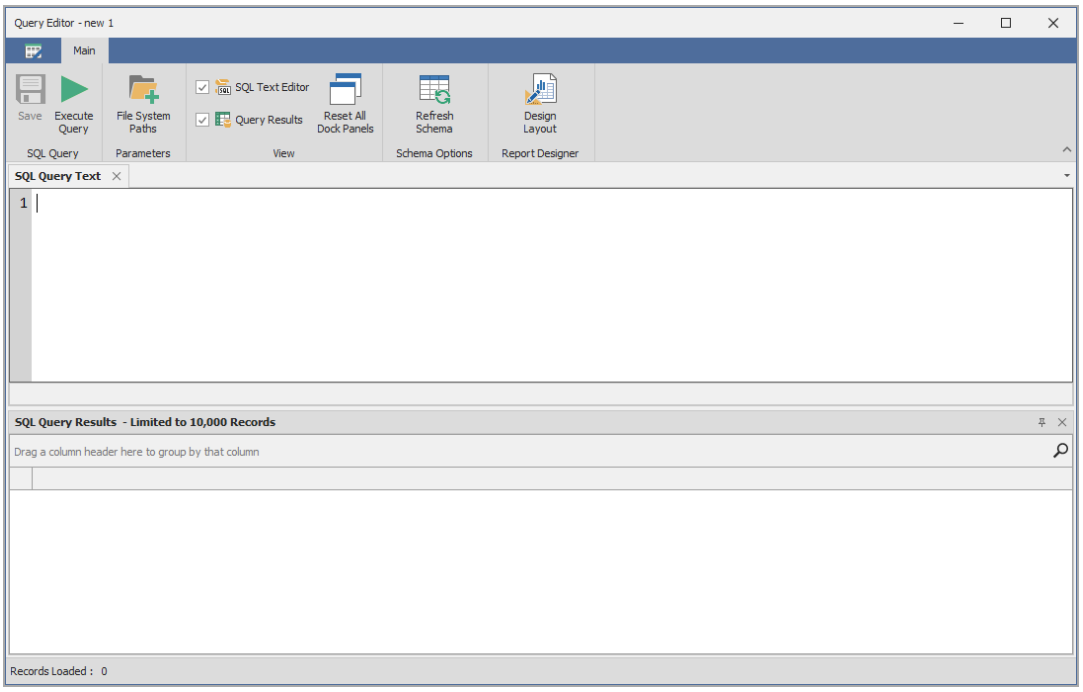


4. Enter the login credentials and click *Login*.



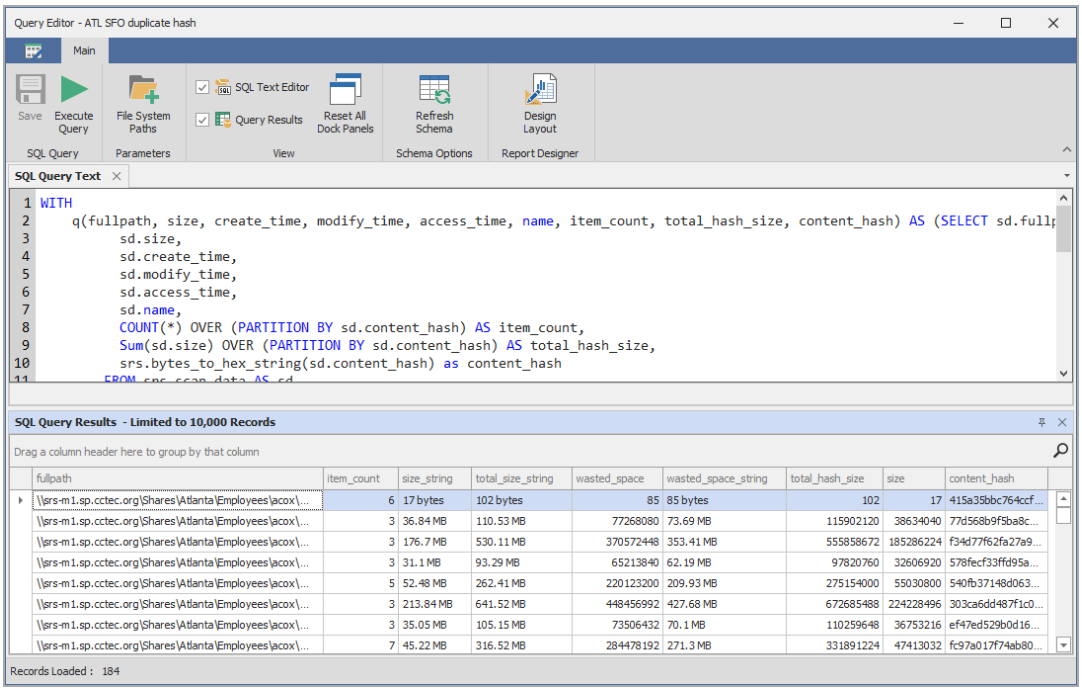
   All of your saved Custom Query reports are listed.

5. Click *New Custom Query*, give it a name, then click *Create*.

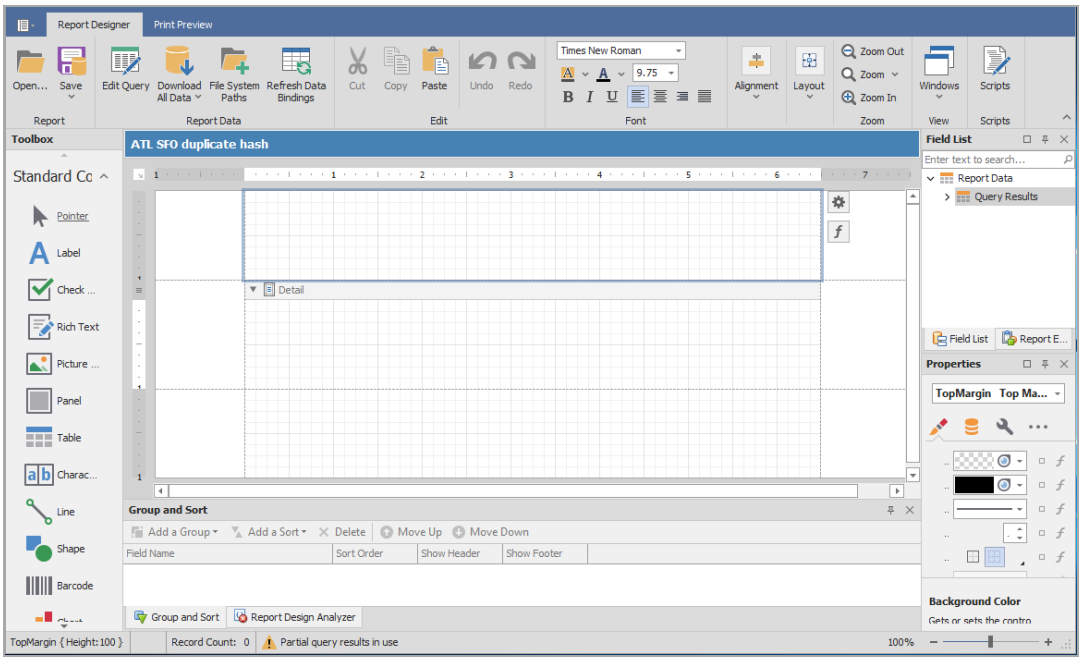


6. From the text editor you used in Step 2, copy the custom query and paste it into the Query Editor.
7. In the line beginning with `WHERE`, edit the UNC paths so that they are specific to the content file hashed shares on which you want to report.

   The custom query only includes two paths so if you want more, extend the line to include more paths by adding `srs.path_hash('\\server\share\path')` to the comma delimited `sd.fullpath_hash IN` portion of the where clause for each desired path.
8. (Conditional) At the bottom of the custom query, modify the `q.item_count` and `q.size` settings to the minimum number of duplicates and file sizes (in bytes), respectively, to include in the report.
9. Click *Execute* to see a preview of the report data.



10. Click *Save*.
11. Click *Design Layout*.



12. Click *Open*.
13. Locate the `.repx` file that you saved and unzipped in Step 2 and click *Open*.

The layout template appears in the Report Designer.

14. Click *Download All Data*.
15. In the subsequent dialog box, click *Yes*.

    This runs the query in the database and loads data into the report template.
16. Click *Print Preview* to review the report findings.

    Note how the hashes are listed with a total number for each and the location of each, meaning the total number of duplicate files and their locations.
17. Save the report by doing one of the following:
    - From the *Export To* drop-down menu, select the file type you want to save the report layout to.
    - Click *Save Report* to save the report as a `.prnx` file that you can open in the Report Viewer and if you want later, export the report to the desired file type.

4.2 - Microsoft 365 Reports

Once Agent365 scans the data and associated permissions for Microsoft 365 file repositories, you can use the pre-built custom queries and associated report layouts in https://filequerycookbook.com to generate reports.

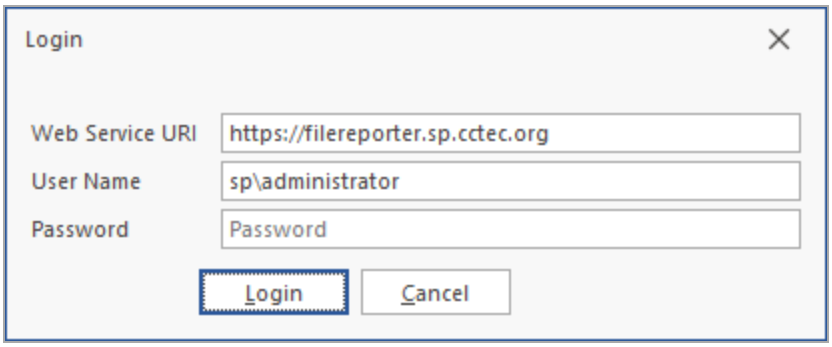
4.2.1 - Determining Prerequisites

- Install and configure Agent365 – see Agent365 in the *File Reporter 26.2 Installation Guide* for details.
- Scan the tenant – see Tenants in the *File Reporter 26.2 Administration Guide* for details.
- Install the Client Tools – see Installing the Client Tools in the *File Reporter 26.2 Client Tools Guide* for details.

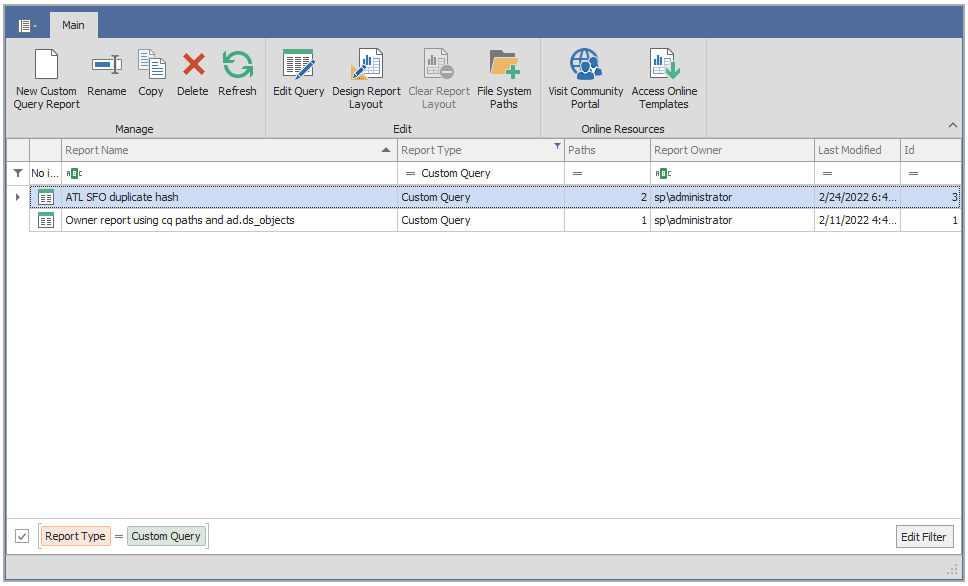
4.2.2 - Designing the Report

The Client Tools include the Report Designer application used in these procedures.

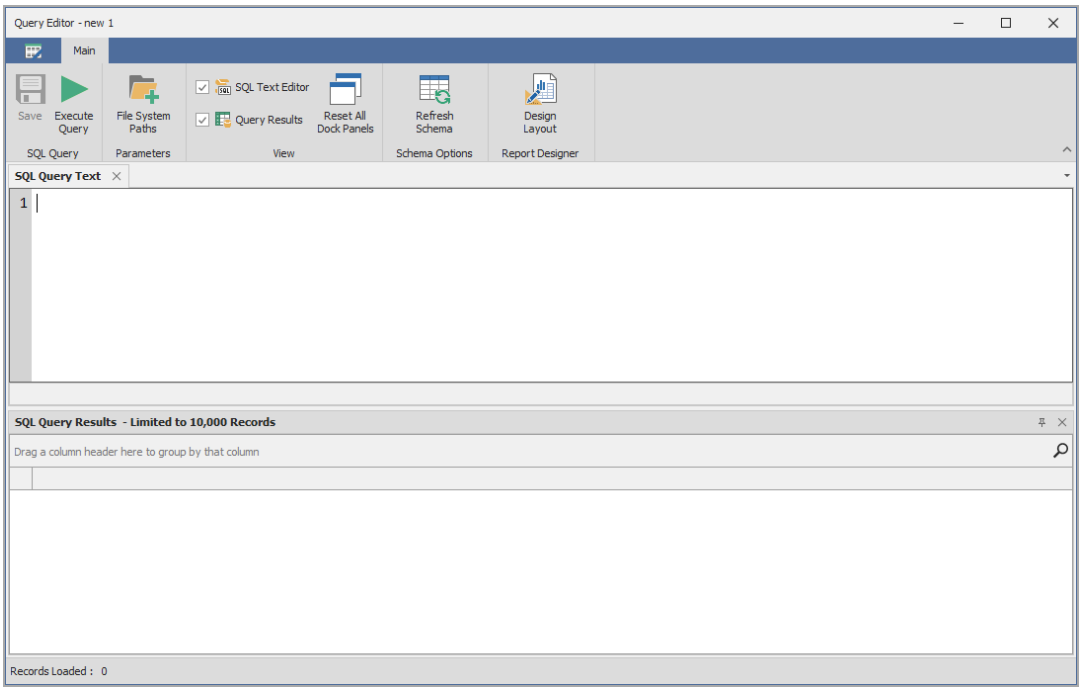
1. Locate and download one of the custom queries and associated reports for Microsoft 365 in the File Query Cookbook (https://filequerycookbook.com). The file is saved as a zip archive.
2. the downloaded file and open the `.sql` file in a text editor. You will eventually paste this custom query into the Query Editor.
3. Launch the File Reporter 26.2 Report Designer in the *Start* menu.



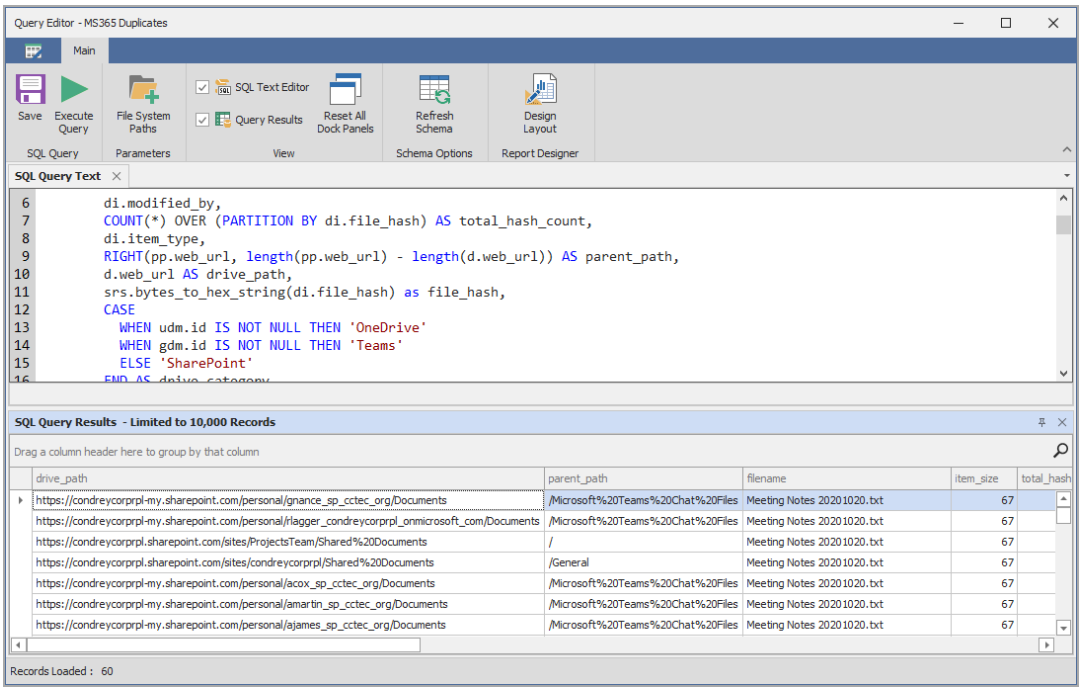
4. Enter your login credentials and click *Login* to open a list of your saved Custom Query reports.



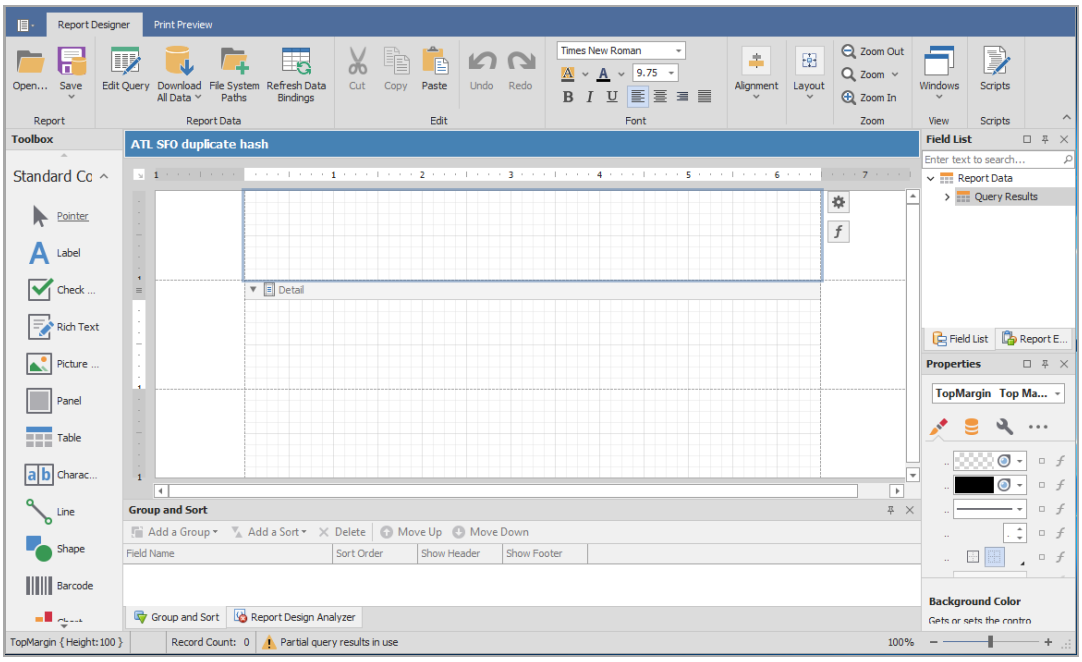
5. Click *New Custom Query*, enter a descriptive name, then click *Create* to launch the Report Designer Query Editor.



6. Copy the custom query you opened in the text editor (see *the downloaded file and open the .sql file in a text editor. You will eventually paste this custom query into the Query Editor. (page 33)*) and paste it into the Query Editor.
7. (Conditional) If there are target paths or other modifications that need to be made for your environment, follow the procedures for the recipe.
8. Click *Execute* to open a preview of the report data in the bottom portion of the editor.



9. Click *Save*.
10. Click *Design Layout*.



11. Click *Open*.

12. Locate the `.repx` file you saved and unzipped in *the downloaded file and open the .sql file in a text editor. You will eventually paste this custom query into the Query Editor. (page 33)* , and click *Open*. The layout template now appears in the Report Designer.
13. Click *Download All Data*.
14. Click *Yes* in the dialog box that opens to run the query in the database and load the data into the report template.
15. Click *Print Preview* to review the report findings.
16. Save the report by performing either of the following procedures:
    - Select the desired file type to save the report in the *Export To* drop-down menu.
    - Click *Save Report* to save the report as a `.prnx` file that you can open in the Report Viewer and export it to the desired file type.

4.3 - Active Directory Identity Enrichment

You can provide extended data for identities in Custom Query reports or create identity reports for security principals in Active Directory.

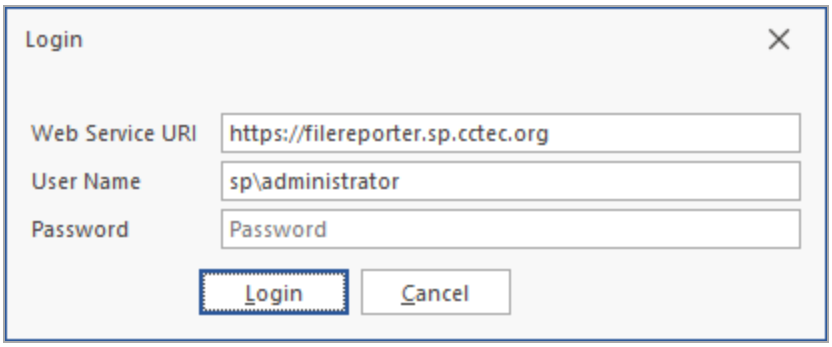
4.3.1 - Determining Prerequisites

- File Reporter collects Active Directory identity data automatically once per day by default – see Active Directory Identity Scans in the *File Reporter 26.2 Administration Guide* for details on running a collection manually.
- Decide whether to extend an existing Custom Query file system metadata or permissions report, or report just on Active Directory identities themselves.
    - If extending an existing Custom Query report, determine whether that report data already includes Security Identifiers (SIDs) or GUIDs of the owner or permissions trustee.
    - If reporting solely on Active Directory identities, determine which of the extended attributes to include in the report – see the table and view definitions for *ad.domains (page 43)*, *ad.ds_objects (page 44)*, and *ad.ds_objects_view (page 113)* for details on available attributes.

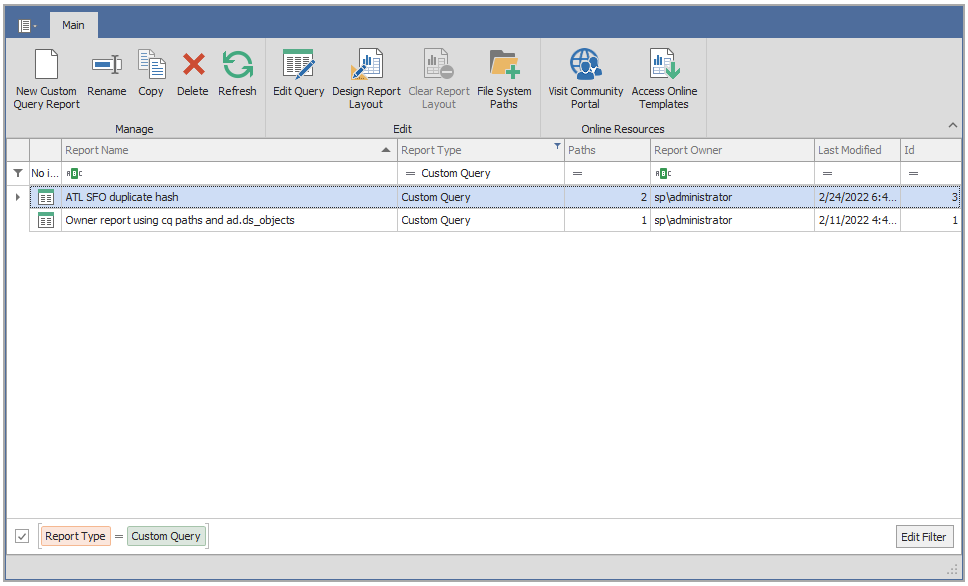
4.3.2 - Designing the Report

The following example extends a "Direct User Assignment" Custom Query report, which identifies user accounts that have been assigned permissions directly to folders (as opposed to group membership), and shows a summary of the count of direct permissions per user by share path.

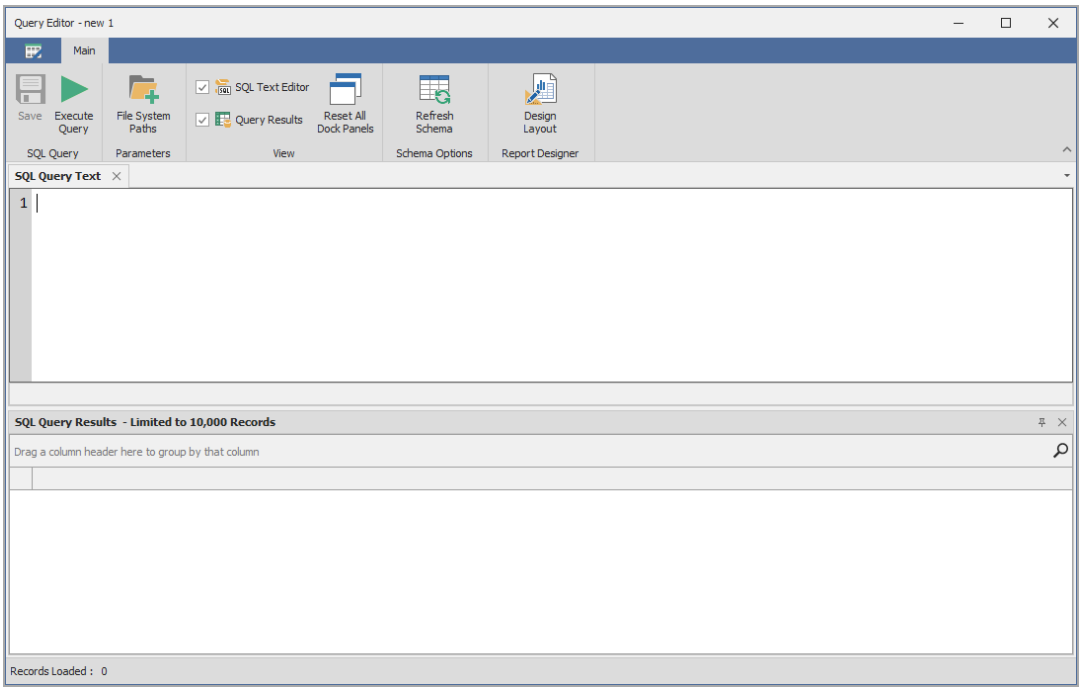
1. Launch the File Reporter26.2 Report Designer in the *Start* menu.



2. Enter your login credentials and click *Login* to open a list of your saved Custom Query reports.



3. Click *New Custom Query*, enter a descriptive name, and click *Create* to launch the Report Designer Query Editor.

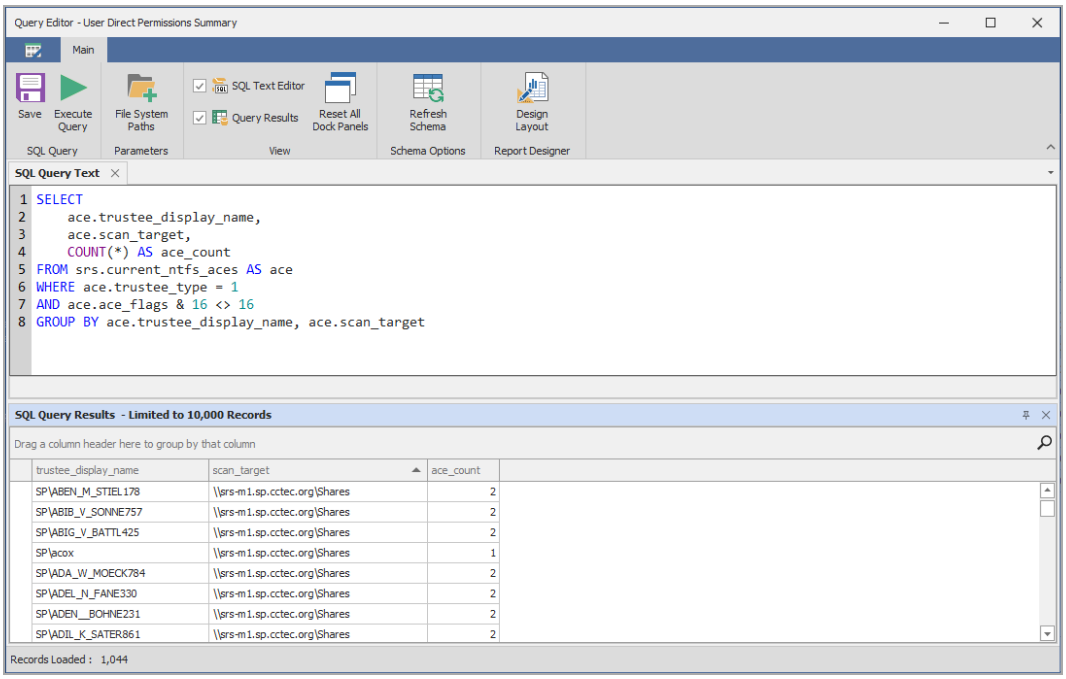


4. Enter the following SQL statements into the Query Editor:

Basic Query - User Direct Permissions Summary

```
SELECT
   ace.trustee_display_name,
   ace.scan_target,
   COUNT(*) AS ace_count
FROM srs.current_ntfs_aces AS ace
WHERE ace.trustee_type = 1
   AND ace.ace_flags & 16 <> 16
GROUP BY
   ace.trustee_display_name,
   ace.scan_target
```

5. Click *Execute* to open a preview of the report data. This query will produce a result similar to the following:

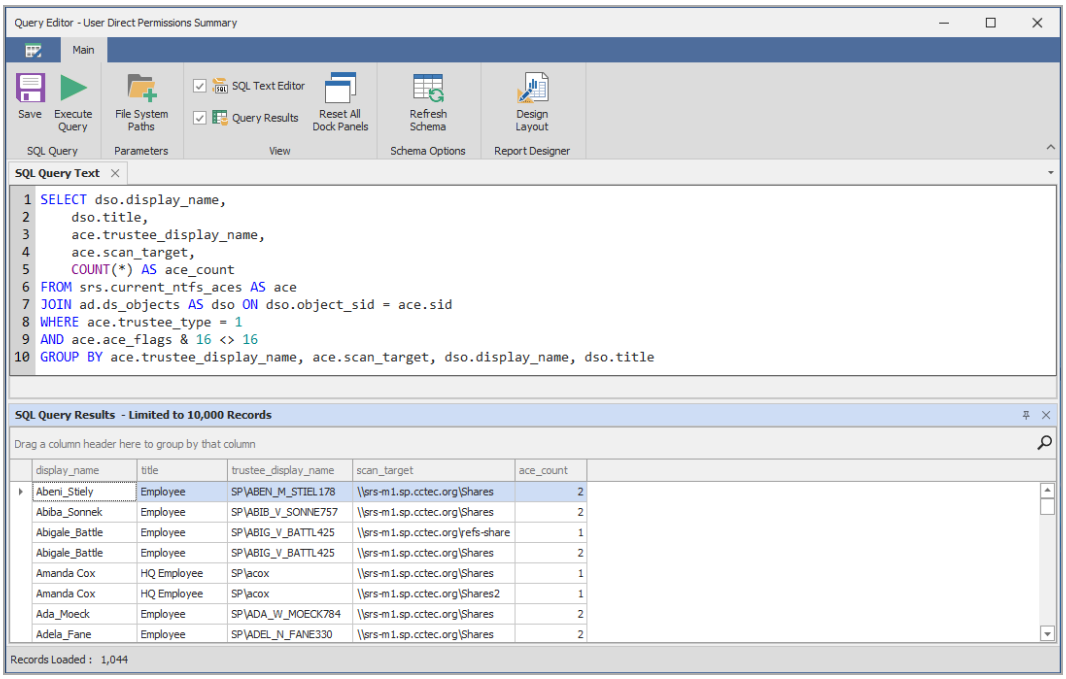


6. Click *Save* to save the SQL you've entered to this point.
7. Augment the data by joining with the *ad.ds_objects* table to include the Active Directory user *display_name* and *title* fields.

Enhanced Query - User Direct Permissions Summary

```
SELECT
  dso.display_name,
  dso.title,
  ace.trustee_display_name,
  ace.scan_target,
  COUNT(*) AS ace_count
FROM srs.current_ntfs_aces AS ace
JOIN ad.ds_objects AS dso
 ON dso.object_sid = ace.sid
WHERE ace.trustee_type = 1
 AND ace.ace_flags & 16 <> 16
GROUP BY
  ace.trustee_display_name,
  ace.scan_target,
  dso.display_name,
  dso.title
```

8. Click Execute to see the updated results, including the *title* and *display_name* fields.


Query Editor - User Direct Permissions Summary
Main
Save
Execute Query
SQL Query
File System Paths
Parameters
SQL Text Editor
Query Results
Reset All Dock Panels
View
Refresh Schema
Schema Options
Design Layout
Report Designer
SQL Query Text
1 SELECT dso.display_name,
2 dso.title,
3 ace.trustee_display_name,
4 ace.scan_target,
5 COUNT(*) AS ace_count
6 FROM srs.current_ntfs_aces AS ace
7 JOIN ad.ds_objects AS dso ON dso.object_sid = ace.sid
8 WHERE ace.trustee_type = 1
9 AND ace.ace_flags & 16 <> 16
10 GROUP BY ace.trustee_display_name, ace.scan_target, dso.display_name, dso.title
SQL Query Results - Limited to 10,000 Records
Drag a column header here to group by that column
display_name | title | trustee_display_name | scan_target | ace_count
Abeni_Stiely | Employee | SP\ABEN_M_STIEL178 | \\srs-m1.sp.cctec.org\Shares | 2
Abiba_Sonnek | Employee | SP\ABIB_V_SONNE757 | \\srs-m1.sp.cctec.org\Shares | 2
Abigale_Battle | Employee | SP\ABIG_V_BATTL425 | \\srs-m1.sp.cctec.org\refs-share | 1
Abigale_Battle | Employee | SP\ABIG_V_BATTL425 | \\srs-m1.sp.cctec.org\Shares | 2
Amanda Cox | HQ Employee | SP\acox | \\srs-m1.sp.cctec.org\Shares | 1
Amanda Cox | HQ Employee | SP\acox | \\srs-m1.sp.cctec.org\Shares2 | 1
Ada_Moeck | Employee | SP\ADA_W_MOECK784 | \\srs-m1.sp.cctec.org\Shares | 2
Adela_Fane | Employee | SP\ADEL_N_FANE330 | \\srs-m1.sp.cctec.org\Shares | 2
Records Loaded : 1,044

5 - Schema Reference

5.1 - Tables

ad.domains

Column Name	SQL Server	PostgreSQL	Notes
id	bigint		Primary key
db_last_update	datetime2(3)	timestamp without time zone	Last update time for this entry in the database
domain_netbios	nvarchar(15)	varchar(15)	Domain NetBIOS name
domain_dns	nvarchar(256)	varchar(256)	Domain DNS name
domain_sid	varbinary(68)	bytea	Domain security identifier
forest_dns	nvarchar(2560	varchar(256)	Forest DNS name

ad.ds_objects

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
db_domain_sid	varbinary(68)	bytea	SID of the domain itself
db_last_update	datetime2(3)	timestamp	Last update time for this entry in the database
object_guid	binary(16)	bytea	Object's GUID
object_ category	nvarchar(256)	varchar(256)	Using LDAP display name, not FDN.
object_class	nvarchar(256)	varchar(256)	Only includes structural class value from this multi-value attribute.
object_sid	varbinary(68)	bytea	Object's Security Identifier
dn	nvarchar (max)	text	Distinguished name
upn	nvarchar (1024)	varchar(1024)	User principal name
sam_account_ name	nvarchar(256)	varchar(256)	SAM account name
sam_account_ type	integer	integer	See https://docs.microsoft.com/en-us/windows/win32/adschema/a-samaccounttype for details. Enum values: 0x00000000 - Domain 0x10000000 - Group 0x10000001 - Non-security Group object 0x20000000 - Alias object

Column Name	SQL Server	PostgreSQL	Notes
			0x20000001 - Non-security Alias object 0x30000000 - Normal User account 0x30000001 - Machine (computer) account 0x30000002 - Trust account 0x40000000 - APP_BASIC Group 0x40000001 - APP_QUERY Group
sam_principal_name	nvarchar(256)	varchar(256)	NetBIOS\SamAccountName. From msDS-PrincipalName. Note that the NetBIOS name here may be different from the associated domain NetBIOS name where this account was scanned. This is especially true for domain Builtin* accounts and foreign security principals.
display_name	nvarchar(256)	varchar(256)	
uac_flags	integer	integer	Combines both userAccessControl and msDs-User-Account-Control-Computed attribute values into a single flag. See the following for details: • https://docs.microsoft.com/en-us/windows/win32/adschema/a-useraccountcontrol • https://docs.microsoft.com/en-us/windows/win32/adschema/a-msds-user-account-control-computed

Column Name	SQL Server	PostgreSQL	Notes
			Flags values: 0x00000001 - Logon script is executed 0x00000002 - User Account disabled 0x00000008 - Home directory required 0x00000010 - Account currently locked out 0x00000020 - No password required 0x00000040 - User cannot change password 0x00000080 - User can send encrypted password 0x00000100 - Temporary duplicate account 0x00000200 - Normal account - typical user 0x00000800 - Inter-domain trust account 0x00001000 - Computer (Workstation / Member Server) account 0x00002000 - Domain controller computer account 0x00010000 - Password does not expire 0x00020000 - Majority Node Set (MNS) logon account 0x00040000 - Smart card required for logon 0x00080000 - Service account trusted for Kerberos delegation 0x00100000 - Account not allowed trust for delegation

Column Name	SQL Server	PostgreSQL	Notes
			0x00200000 - Account can only use DES keys 0x00400000 - Account does not require Kerberos pre-authentication for logon 0x00800000 - User password has expired 0x01000000 - Account enabled for delegation 0x04000000 - Partial secrets account 0x08000000 - Account can only use Use AES keys
account_ expires	datetime2(0)	timestamp	
create_ timestamp	datetime2(0)	timestamp	
description	nvarchar (1024)	varchar(1024)	Only uses first value of this multi-value attribute
mail	nvarchar(256)	varchar(256)	
given_name	nvarchar(64)	varchar(64)	
surname	nvarchar(64)	varchar(64)	
last_logon_ timestamp	datetime2(0)	timestamp	NOTE: This attribute only has 14-day granularity. See: https://docs.microsoft.com/en-us/windows/win32/adschema/a-lastlogontimestamp
department	nvarchar(64)	varchar(64)	

Column Name	SQL Server	PostgreSQL	Notes
title	nvarchar(128)	varchar(128)	
primary_group_sid	varbinary(68)	bytea	SID of referenced object
managed_by_guid	binary(16)	bytea	GUID of referenced DS object
manager_guid	binary(16)	bytea	GUID of referenced DS object
group_type	integer	integer	See https://docs.microsoft.com/en-us/windows/win32/adschema/a-grouptype for details. Flags: 0x01 - System created group 0x02 - Global group 0x04 - Domain Local group 0x08 - Universal group 0x10 - APP_BASIC group for Windows Server Authorization Manager 0x20 - APP_QUERY group for Windows Server Authorization Manager 0x80000000 - Security Group. If not set, then a Distribution Group
dns_host_name	nvarchar (2048)	varchar(2048)	Applies to Computer objects

srs.analysis.file_scan_entries

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
scan_time	datetime2(3)	timestamp	Time when file content was scanned
fullpath	nvarchar(max)	text	Full UNC path to the file
fullpath_hash	binary(20)	bytea	SHA-1 hash of lowercase fullpath
content_hash	binary(32)	bytea	SHA-2 hash of file content
size	bigint	bigint	File size
modify_time	datetime2(2)	timestamp	Last write time of file
classification	nvarchar(64)	varchar(64)	Classification name
category	nvarchar(64)	varchar(64)	Category name
search_pattern_name	nvarchar(64)	varchar(64)	Search pattern name
search_pattern_string	nvarchar(1024)	varchar(1024)	Search pattern string
match_count	int	int	Number of matches for Search Pattern on this path
match_confidence	int	int	1 = Low 2 = Medium 3 = High
job_id	int	int	File content scan job ID
job_definition	nvarchar(64)	varchar(64)	Job definition name

Column Name	SQL Server	PostgreSQL	Notes
status_code	int	int	Processing status code for this file entry

ms365.drive_item_types

Column Name	SQL Server	PostgreSQL	Notes
item_type	int	int	0 = unknown 1 = file 2 = folder 3 = remote_item
item_type_name	nvarchar(32)	varchar(32)	Item type description

ms365.drive_items

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
scan_id	bigint	bigint	Reference to primary key in ms365.drive_scans
drive_id	bigint	bigint	Reference to associated drive in ms365.drives
ms365_id	nvarchar (256)	varchar(256)	Unique ID provided by MS GraphAPI
ms365_drive_id	nvarchar (256)	varchar(256)	Unique ID provided by MS GraphAPI for the associated drive
ms365_parent_id	nvarchar (256)	varchar(256)	Unique ID provided by MS GraphAPI for parent path
created_by	nvarchar (256)	varchar(256)	Unique ID provided by MS GraphAPI for the associated identity
created_by_name	nvarchar (256)	varchar(256)	Display name of the "created_by" account
create_time	datetime2(3)	timestamp	Create time for entry
item_type	integer	integer	Note: Only one of these values is set as a "primary" value for this entry as opposed to the item_facets column 0 = unknown 1 = file 2 = folder 4 = package 8 = remote item
item_facets	integer	integer	Note: All applicable flags are set for this value, as opposed to the item_type column

Column Name	SQL Server	PostgreSQL	Notes
			0 = none 1 = file 2 = folder 4 = package 8 = remote item
file_hash	varbinary(64)	varchar(64)	Files only - QuickXorHash of entry See https://docs.microsoft.com/en-us/graph/api/resources/hashes?view=graph-rest-1.0
child_count	bigint	bigint	Folders only - number of child entries in the folder Only includes immediate children, not recursive.
modified_by	nvarchar (256)	varchar(256)	Unique ID provided by MS GraphAPI for the associated identity
modified_by_name	nvarchar (256)	varchar(256)	Display name of the "modified_by" account
modify_time	datetime2(3)	timestamp	Last modified time
name	nvarchar (400)	varchar(400)	Name of entry. See https://support.microsoft.com/en-us/office/restrictions-and-limitations-in-onedrive-and-sharepoint-64883a5d-228e-48f5-b3d2-eb39e07630fa#filenamepathlengths
file_extension	nvarchar(32)	varchar(32)	File name extension
size	bigint	bigint	Size in bytes
web_url	nvarchar (max)	text	Full path to item

Column Name	SQL Server	PostgreSQL	Notes
web_url_hash	varbinary(32)	bytea	SHA-256 hash of web_url

ms365.drive_scans

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	integer	integer	Reference to primary key in ms365.jobs
drive_id	bigint	bigint	Reference to primary key in ms365.drives
scan_status	integer	integer	0 = Queued 1 = In progress 2 = Completed 3 = Failed 99 = Canceled
scan_state	integer	integer	0 = Pending 1 = Current 99 = Marked for cleanup
delegated_time	datetime2(3)	timestamp	Time at which scan was requested
start_time	datetime2(3)	timestamp	Time when scan started
stop_time	datetime2(3)	timestamp	Time when scan stopped
scan_progress_data	nvarchar(max)	text	JSON data with scan progress details
agent_name	nvarchar(256)	varchar(256)	Name of Agent365 server performing the scan

ms365.drive_scans_history

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	int	int	Reference to primary key in ms365.jobs
scan_id	bigint	bigint	Reference to primary key in ms365.drive_scans
start_time	datetime2(3)	timestamp	Drive scan start time
stop_time	datetime2(3)	timestamp	Drive scan stop time
drive_id	bigint	bigint	Reference to primary key in ms365.drives
drive_name	nvarchar(256)	varchar(256)	Drive name
web_url	nvarchar(max)	text	Full path to drive
ms365_drive_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
scan_progress_status	nvarchar(max)	text	JSON data with scan progress details
agent_name	nvarchar(256)	varchar(256)	Name of Agent365 server that performed the scan
scan_status	int	int	0 = Queued 1 = In progress 2 = Completed 3 = Failed 99 = Canceled
scan_state	int	int	0 = Pending 1 = Current

Column Name	SQL Server	PostgreSQL	Notes
			99 = Marked for cleanup
result_string	nvarchar(max)	text	Success or error message

ms365.drives

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	int	int	Reference to primary key in ms365.jobs
tenant_id	int	int	Reference to primary key in ms365.tenants table
site_id	bigint	bigint	Reference to primary key in ms365.sites table
last_update	datetime2(3)	timestamp	Last update time for database entry
ms365_id	nvarchar (256)	varchar(256)	Unique id provided by MS GraphAPI
name	nvarchar (256)	varchar(256)	Drive name
ms365_ owner_id	nvarchar (256)	varchar(256)	Unique id provided by MS GraphAPI
quota	nvarchar (256)	varchar(256)	JSON data including quota details
web_url	nvarchar (max)	text	Full web path to drive
drive_type	nvarchar(64)	varchar(64)	Known values in MS GraphAPI include • business • documentLibrary See: https://docs.microsoft.com/en-us/graph/api/resources/drive?view=graph-rest-1.0

ms365.group_drives

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	int	int	Reference to primary key in ms365.jobs
tenant_id	int	int	Reference to primary key in ms365.tenants
last_update	datetime2(3)	timestamp	Last update time for database entry
ms365_group_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated group
ms365_drive_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated drive

ms365.group_member_types

Column Name	SQL Server	PostgreSQL	Notes
member_type	int	int	0 = direct 1 = transitive
member_type_name	nvarchar(32)	varchar(32)	Member type description

ms365.group_members

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	int	int	Reference to primary key in ms365.jobs
tenant_id	int	int	Reference to primary key in ms365.tenants
last_update	datetime2(3)	timestamp	Last update time for database entry
ms365_group_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated group
ms365_member_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated member
member_type	int	int	0 = direct 1 = transitive

ms365.group_owners

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	int	int	Reference to primary key in ms365.jobs
tenant_id	int	int	Reference to primary key in ms365.tenants
last_update	datetime2(3)	timestamp	Last update time for database entry
ms365_group_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated group
ms365_owner_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated owner

ms365.group_sites

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	int	int	Reference to primary key in ms365.jobs
tenant_id	int	int	Reference to primary key in ms365.tenants
last_update	datetime2(3)	timestamp	Last update time for database entry
ms365_group_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated group
ms365_site_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated SharePoint site

ms365.groups

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	int	int	Reference to primary key in ms365.jobs
tenant_id	int	int	Reference to primary key in ms365.tenants
last_update	datetime2(3)	timestamp	Last update time for database entry
ms365_id	nvarchar (256)	varchar(256)	Unique id provided by MS GraphAPI
display_name	nvarchar (256)	varchar(256)	Friendly name of group
email	nvarchar (256)	varchar(256)	Email address
group_types	nvarchar(64)	varchar(64)	One or more of the following from MS GraphAPI: • Unified • DynamicMembership • [empty string] See: https://docs.microsoft.com/en-us/graph/api/resources/group?view=graph-rest-1.0
onprem_sid	varbinary(68)	bytea	On-premises Security Identifier (SID)
onprem_ dnsdomain	nvarchar (256)	varchar(256)	On-premises DNS domain
onprem_ netbios	nvarchar (256)	varchar(256)	On-premises NetBIOS domain
onprem_ samaccount	nvarchar (256)	varchar(256)	On-premises SAM Account Name

ms365.identity_types

Column Name	SQL Server	PostgreSQL	Notes
identity_type	int	int	0 = unknown 1 = user 2 = group 3 = device 4 = application
identity_type_name	nvarchar(32)	varchar(32)	Identity type description

ms365.jobs

Column Name	SQL Server	PostgreSQL	Notes
id	int	int	Primary key
tenant_id	int	int	Reference to primary key in ms365.tenants
start_time	datetime2(3)	timestamp	Time job started
stop_time	datetime2(3)	timestamp	Time job stopped
job_status	int	int	0 = Queued 1 = In progress 2 = Completed 3 = Failed 99 = Canceled
job_progress_data	nvarchar(max)	text	JSON data with job progress details
agent_name	nvarchar(256)	varchar(256)	Agent365 server performing the scan

ms365.jobs_history

Column Name	SQL Server	PostgreSQL	Notes
id	int	int	Primary key
job_id	int	int	Reference to primary key in ms365.jobs
tenant_id	int	int	Reference to primary key in ms365.tenants
tenant_name	nvarchar(256)	varchar(256)	Associated *.onmicrosoft.com tenant name
start_time	datetime2(3)	timestamp	Time when job started
stop_time	datetime2(3)	timestamp	Time when job stopped
job_status	int	int	0 = Queued 1 = In progress 2 = Completed 3 = Failed 99 = Canceled
result_string	nvarchar(1024)	varchar(1024)	Success or failure message
job_progress_data	nvarchar(max)	text	JSON data with job progress details
agent_name	nvarchar(256)	varchar(256)	Agent365 server performing the scan

ms365.permissions

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
scan_id	bigint	bigint	Reference to primary key in ms365.drive_scans
site_collection_id	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
drive_item_id	bigint	bigint	Reference to primary key in ms365.drive_items
ms365_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
expire_time	datetime2(3)	timestamp	Timestamp when link expires
is_inherited	bit	boolean	true = inherited false = not inherited
has_password	bit	boolean	This currently applies only to Anonymous sharing links
grantedto_ms365_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated trustee
grantedto_type	integer	integer	0 = unknown 1 = user 2 = group 3 = device 4 = application
grantedto_sp_user_id	integer	integer	Reference to an associated SharePoint site collection's user account
grantedto_sp_group_id	integer	integer	Reference to an associated SharePoint site collection's group account

Column Name	SQL Server	PostgreSQL	Notes
grantedto_sp_login_name	nvarchar (256)	varchar(256)	SharePoint-specific login name for the trustee
grantedto_display_name	nvarchar (256)	varchar(256)	Friendly name of trustee
grantedto_email	nvarchar (256)	varchar(256)	Email address of trustee
invite_email	nvarchar (256)	varchar(256)	Email address of recipient (trustee)
invite_sentby_ms365_id	nvarchar (256)	varchar(256)	Unique id provided by MS GraphAPI for the associated sender
invite_sentby_display_name	nvarchar (256)	varchar(256)	Friendly name of sender
invite_signin_required	bit	boolean	true = sign-in required false = sign-in not required
link_app_display_name	nvarchar (256)	varchar(256)	Friendly name of application
link_app_ms365_id	nvarchar (256)	varchar(256)	Unique id provided by MS GraphAPI for the associated application
link_type	nvarchar(32)	varchar(32)	One of: • view • edit See: https://docs.microsoft.com/en-us/graph/api/resources/sharinglink?view=graph-rest-1.0
link_scope	nvarchar(32)	varchar(32)	One of the following from MS GraphAPI:

Column Name	SQL Server	PostgreSQL	Notes
			• anonymous • organization See : https://docs.microsoft.com/en-us/graph/api/resources/sharinglink?view=graph-rest-1.0
link_prevents_download	bit	boolean	true = view only (download not allowed)
roles	nvarchar(128)	varchar(128)	One of the following from MS GraphAPI: • read • write • owner See: https://docs.microsoft.com/en-us/graph/api/resources/permission?view=graph-rest-1.0

ms365.sharing_link_members

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
permission_id	bigint	bigint	Reference to primary key in ms365.permissions
scan_id	bigint	bigint	Reference to primary key in ms365.drive_scans
site_collection_id	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
ms365_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated member
member_type	integer	integer	0 - Direct membership 1 - Transitive (nested membership)
display_name	nvarchar(256)	varchar(256)	Friendly name of member
email	nvarchar(256)	varchar(256)	Email address of member
sp_group_id	integer	integer	Reference to an associated SharePoint site collection's group account
sp_user_id	integer	integer	Reference to an associated SharePoint site collection's user account

Column Name	SQL Server	PostgreSQL	Notes
sp_login_name	nvarchar(256)	varchar(256)	SharePoint-specific login name for the member
sp_display_name	nvarchar(256)	varchar(256)	Friendly name of member's associated SharePoint account

ms365.sites

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	int	int	Reference to primary key in ms365.jobs
tenant_id	int	int	Reference to primary key in ms365.tenants
site_collection_id	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
last_update	datetime2(3)	timestamp	Last update time for database entry
ms365_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
ms365_parent_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated parent site
display_name	nvarchar(256)	varchar(256)	Friendly name of SharePoint site
name	nvarchar(256)	varchar(256)	Site name
is_root	bit	boolean	true = root site (no parent sites) false = child site
web_url	nvarchar(max)	text	Full path to SharePoint site

ms365.sp_base_permissions

Column Name	SQL Server	PostgreSQL	Notes
flag	bigint	bigint	Base permissions flag value
name	nvarchar(64)	varchar(64)	Flag entry name
description	nvarchar(1024)	varchar(1024)	Flag entry description

This is a pre-populated lookup table.

Values are derived from SharePoint client and server .NET APIs.

See https://docs.microsoft.com/en-us/previous-versions/office/sharepoint-server/ee536458(v=office.15) and https://docs.microsoft.com/en-us/previous-versions/office/sharepoint-server/ms412690(v=office.15).

ms365.sp_group_members

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	integer	integer	Reference to primary key in ms365.jobs
tenant_id	integer	integer	Reference to primary key in ms365.tenants
last_update	datetime2(3)	timestamp	Last update time for database entry
site_collection_id	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
sp_group_id	integer	integer	Reference to an associated SharePoint site collection's group account
sp_member_id	integer	integer	Reference to an associated SharePoint site collection's user account
ms365_member_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated member

ms365.sp_groups

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	integer	integer	Reference to primary key in ms365.jobs
tenant_id	integer	integer	Reference to primary key in ms365.tenants
site_collection_id	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
sp_id	integer	integer	SharePoint ID for this entry, unique per site collection
last_update	datetime2(3)	timestamp	Last update time for database entry
login_name	nvarchar(256)	varchar(256)	SharePoint account name for this group
title	nvarchar(256)	varchar(256)	Group's title
description	nvarchar(1024)	varchar(1024)	Group's description
is_hidden	bit	boolean	Flag indicating whether this is a hidden group

ms365.sp_permission_levels

Column Nam	SQL Server	PostgreSQl	Notes
id	bigint	bigint	Primary key
job_id	integer	integer	Reference to primary key in ms365.jobs
tenant_id	integer	integer	Reference to primary key in ms365.tenants
site_ collection_ id	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
sp_id	integer	integer	SharePoint ID for this entry, unique per site collection
name	nvarchar (256)	varchar (256)	Name of Permission Level (role)
description	nvarchar (1024)	varchar (1024)	Description for this Permission Level
base_ permission s	bigint	bigint	Flags value indicating the underlying permissions this Permission Level (Role) defines Query or join with the descriptions table ms365.sp_base_permissions. See • https://docs.microsoft.com/en-us/previous-versions/office/sharepoint-server/ee536458(v=office.15) • https://docs.microsoft.com/en-us/previous-versions/office/sharepoint-server/ms412690(v=office.15)
role_type	integer	integer	0 - None 1 - Guest 2 - Reader 3 - Contributor

Column Nam	SQL Server	PostgreSQL	Notes
			4 - Web Designer 5 - Administrator 6 - Editor 7 - Reviewer 8 - Restricted Reader 9 - Restricted Guest 255 - System See https://docs.microsoft.com/en-us/dotnet/api/microsoft.sharepoint.client.roletype?view=sharepoint-csom
is_hidden	bit	boolean	Indicates whether this is a hidden role

ms365.sp_permissions

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
scan_id	bigint	bigint	Reference to primary key in ms365.drive_scans
site_collection_id	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
drive_item_id	bigint	bigint	Reference to primary key in ms365.drive_items
sp_user_id	integer	integer	Reference to an associated SharePoint site collection's user account
sp_group_id	integer	integer	Reference to an associated SharePoint site collection's group account
sp_login_name	nvarchar(256)	varchar(256)	SharePoint account name for the trustee
sp_display_name	nvarchar(256)	varchar(256)	Display name for the trustee
sp_permission_level_id	integer	integer	Reference to primary key in ms365.sp_permission_levels
is_inherited	bit	boolean	Flag indicating whether this is an inherited permission

ms365.sp_site_permissions

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	integer	integer	Reference to primary key in ms365.jobs
site_id	bigint	bigint	Reference to primary key in ms365.sites for the associated site
site_collection_id	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
drive_item_id	bigint	bigint	Reference to primary key in ms365.drive_items
sp_user_id	integer	integer	Reference to an associated SharePoint site collection's user account
sp_group_id	integer	integer	Reference to an associated SharePoint site collection's group account
sp_login_name	nvarchar(256)	varchar(256)	SharePoint account name for the trustee
sp_display_name	nvarchar(256)	varchar(256)	Display name for the trustee
sp_permission_level_id	integer	integer	Reference to primary key in ms365.sp_permission_levels
is_inherited	bit	boolean	Flag indicating whether this is an inherited permission

ms365.sp_users

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	int	int	Reference to primary key in ms365.jobs
tenant_id	int	int	Reference to primary key in ms365.tenants table
site_collection_id	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
ms365_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
sp_id	integer	integer	SharePoint ID for this entry, unique per site collection
last_update	datetime2(3)	timestamp	Last update time for database entry
login_name	nvarchar(256)	varchar(256)	SharePoint account name for this user
upn	nvarchar(256)	varchar(256)	User principal name
email	nvarchar(256)	varchar(256)	User's email address
title	nvarchar(256)	varchar(256)	User's title
principal_type	smallint	smallint	One of the following values as defined by the CSOM 'PrincipalType' enumeration: • 0 : None • 1 : User • 2 : Distribution List • 4 : Security Group

Column Name	SQL Server	PostgreSQL	Notes
			• 8 : SharePoint Group
is_site_admin	bit	boolean	Flag indicating whether this user is assigned as a SharePoint admin for the associated site.
is_hidden	bit	boolean	Flag indicating a hidden account
is_guest	bit	boolean	Flag indicating a guest account
is_email_ authenticated	bit	boolean	Only applies to "external" users with sharing

ms365.team_channels

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	int	int	Reference to primary key in ms365.jobs
tenant_id	int	int	Reference to primary key in ms365.tenants
last_update	datetime2(3)	timestamp	Last update time for database entry
ms365_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
team_id	bigint	bigint	Reference to primary key in ms365.teams
display_name	nvarchar(256)	varchar(256)	Friendly name of channel
web_url	nvarchar(256)	varchar(256)	Full path to channel
ms365_files_folder_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated path
ms365_files_folder_drive_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated path's drive

ms365.teams

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	int	int	Reference to primary key in ms365.jobs
tenant_id	int	int	Reference to primary key in ms365.tenants
last_update	datetime2(3)	timestamp	Last update time for database entry
ms365_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
display_name	nvarchar(256)	varchar(256)	Friendly name of team
visibility	int	int	0 = private 1 = public
web_url	nvarchar(max)	text	Full path to team

ms365.tenants

Column Name	SQL Server	PostgreSQL	Notes
id	int	int	Primary key
tenant_name	nvarchar(256)	varchar(256)	Official registered tenant name ending with '.onmicrosoft.com'
ms365_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
display_name	nvarchar(256)	varchar(256)	Tenant display name
default_name	nvarchar(256)	varchar(256)	Optionally registered DNS name set as the "default" e.g. corp.example.com

ms365.user_drives

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	int	int	Reference to primary key in ms365.jobs
tenant_id	int	int	Reference to primary key in ms365.tenants
last_update	datetime2(3)	timestamp	Last update time for database entry
ms365_user_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated user
ms365_drive_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated drive

ms365.users

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
job_id	int	int	Reference to primary key in ms365.jobs
tenant_id	int	int	Reference to primary key in ms365.tenants
last_update	datetime2(3)	timestamp	Last update time for database entry
ms365_id	nvarchar (256)	varchar(256)	Unique id provided by MS GraphAPI
display_name	nvarchar (256)	varchar(256)	Display name - typically First Last name
upn	nvarchar (1024)	varchar (1024)	User Principal Name
given_name	nvarchar(64)	varchar(64)	First name
surname	nvarchar(64)	varchar(64)	Last name
onprem_sid	varbinary(68)	bytea	On-premises Security Identifier (SID)
onprem_dn	nvarchar (max)	text	On-premises distinguished name
onprem_upn	nvarchar (1024)	varchar (1024)	On-premises User Principal Name
onprem_ dnsdomain	nvarchar (256)	varchar(256)	On-premises DNS domain name
onprem_ samaccount	nvarchar (256)	varchar(256)	On-premises SAM Account Name
onprem_	nvarchar	varchar(256)	Unique id mapping synced on-prem

Column Name	SQL Server	PostgreSQL	Notes
immutable_id	(256)		user to associated MS365 user
account_ enabled	bit	boolean	Account is enabled
user_type	nvarchar(64)	varchar(64)	Known values from MS GraphAPI include: • Member • Guest See: https://docs.microsoft.com/en-us/graph/api/resources/user?view=graph-rest-1.0
creation_type	nvarchar(64)	varchar(64)	Known values from MS GraphAPI include: • [null] • Invitation • LocalAccount • EmailVerified See : https://docs.microsoft.com/en-us/graph/api/resources/user?view=graph-rest-1.0

srs.ad_memberships

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
group_id	integer	integer	Reference to primary key in srs.ad_objects
member_id	integer	integer	Reference to primary key in srs.ad_objects

srs.ad_objects

Column Name	SQL Server	PostgreSQL	Notes
id	integer	integer	Primary key
name	nvarchar(256)	varchar(256)	SAM Account Name
fdn	nvarchar(512)	varchar(512)	Full distinguished object name
domain	nvarchar(256)	varchar(256)	Domain name
guid	binary(16)	bytea	Globally Unique Identifier
sid	varbinary(68)	bytea	Security Identifier
object_type	integer	integer	0 = Unknown / Other 1 = User 2 = Group 3 = Computer
identity_system_id	integer	integer	Reference to primary key of identity_systems table

srs.identity_systems

Column Name	SQL Server	PostgreSQL	Notes
id	integer	integer	Primary key
type	integer	integer	0 = Unknown 1 = Active Directory 3 = Windows Local
name	nvarchar(256)	varchar(256)	One of: AD Forest Root DNS name Member server NetBIOS name Built-in Account Prefix
domain	nvarchar(256)	varchar(256)	AD Forest Root NetBIOS name
proxy_account	nvarchar(256)	varchar(256)	
is_primary	bit	boolean	0 = Not the primary identity system 1 = Primary identity system for authentication
is_managed	bit	boolean	0 = Not managed (member server, built-in domain, etc.) 1 = Managed, configured system
last_modified	datetime2(0)	timestamp	

srs.ntfs_aces

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
scan_data_id	bigint	bigint	Reference to scan_data table
flags	smallint	smallint	0x1 = Object Inherit 0x2 = Container Inherit 0x4 = No Propagate 0x8 = Inherit Only 0x10 = Inherited 0x40 = Successful Access 0x80 = Failed Access
ace_type	smallint	smallint	0 = Access Allowed 1 = Access Denied 2 = System Audit 9 = Allowed Callback 10 = Denied Callback 13 = System Audit Callback 17 = System Mandatory Label
access_mask	integer	integer	0x1 = Read Data / List Directory 0x2 = Write Data / Create File 0x4 = Append Data / Create Subdirectory 0x8 = Read Extended Attributes 0x10 = Write Extended Attributes

Column Name	SQL Server	PostgreSQL	Notes
			0x20 = File Execute / Traverse 0x40 = Delete Child 0x80 = Read Attributes 0x100 = Write Attributes 0x10000 = Delete 0x20000 = Read Permissions 0x40000 = Change Permissions 0x80000 = Change Owner 0x100000 = Synchronize 0x1000000 = Access System Security 0x10000000 = Generic All 0x20000000 = Generic Execute 0x40000000 = Generic Write 0x80000000 = Generic Read
sid	varbinary(68)	bytea	Trustee Security Identifier (SID)
index_on_disk	smallint	smallint	Discovered order of this ACE for the associated entry as read from the file system

srs.scan_data

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
scan_id	integer	integer	Reference to primary key in srs.scans
path_type	integer	integer	0 = Unknown 1 = File 2 = Directory 3 = File Symbolic Link 4 = Directory Symbolic Link 5 = Junction 6 = Mount Point 7 = Share 8 = Volume 9 = DFS Link 10 = DFS Folder 11 = DFS Root 12 = HSM Stub 13 = Reparse Point Unknown 17 = Single Instance Storage Stub 18 = Named Stream
is_link	bit	boolean	Flag indicating entry is a link (symlink, hardlink, etc.)
name	nvarchar(256)	varchar(256)	File or directory name
fullpath	nvarchar(max)	text	Full UNC path to the file system entry

Column Name	SQL Server	PostgreSQL	Notes
fullpath_hash	binary(20)	bytea	SHA-1 hash of lowercase fullpath
filename_extension	nvarchar(32)	varchar(32)	Extensions having more than 32 characters are treated as if they have none
owner_id	varbinary(68)	bytea	Security Identifier (SID)
attributes	integer	integer	0x0 = None 0x1 = Read Only 0x2 = Archive 0x4 = System 0x8 = Hidden 0x10 = Directory 0x20 = Compressed 0x40 = Offline 0x80 = NTFS device 0x100 = NTFS Normal 0x200 = NTFS Temporary 0x400 = NTFS Sparse File 0x800 = NTFS Reparse Point 0x1000 = NTFS Not content indexed 0x2000 = NTFS Encrypted 0x4000 = NTFS Virtual
create_time	datetime2(0)	timestamp	
modify_time	datetime2(0)	timestamp	
access_time	datetime2(0)	timestamp	

Column Name	SQL Server	PostgreSQL	Notes
size	bigint	bigint	For files, actual size; for directories, accumulative size of all subordinate files
size_on_disk	bigint	bigint	Assumes typical allocation unit size of 4K
size_compressed	bigint	bigint	Only accurate for NTFS file systems
idx	integer	integer	Scan index; unique per scan
parent_idx	integer	integer	Parent index. Used for hierarchical relation processing
path_depth	integer	integer	Entry depth with respect to the scan target's root path.
ns_left	integer	integer	Nested-set Left index - used for hierarchical relation processing
ns_right	integer	integer	Nested-set Right index - used for hierarchical relation processing
status_code	integer	integer	

srs.scan_directory_data

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
scan_data_id	bigint	bigint	Reference to scan_data table
file_count	integer	integer	Count of all files subordinate to this directory
directory_count	integer	integer	Count of all subdirectories
directory_quota	bigint	bigint	Directory quota for this directory
directory_quota_flags	integer	integer	0 = Unknown 1 = Enforced 2 = Disabled 4 = Incomplete 8 = Rebuilding
child_file_count	integer	integer	Count of all immediately subordinate files
child_link_count	integer	integer	Count of all immediately subordinate links
child_directory_count	integer	integer	Count of all immediately subordinate directories
child_size	bigint	bigint	Size of all immediately subordinate files
child_size_on_disk	bigint	bigint	Size on disk of all immediately subordinate files (assumes 4K allocation size)
child_size_	bigint	bigint	Size on disk of all

Column Name	SQL Server	PostgreSQL	Notes
compressed			immediately subordinate compressed files (only accurate with NTFS)
child_link_size	bigint	bigint	Size of all immediately subordinate links

srs.scan_history

Column Name	SQL Server	PostgreSQL	Notes
id	integer	integer	Primary key
identity_system	nvarchar(256)	text	Identity system associated with this scan target
scan_target	nvarchar(1024)	text	UNC path of scan target
file_size	bigint	bigint	Total aggregate size of all files
file_count	integer	integer	Total count of all files
directory_count	integer	integer	Total count of all directories
scan_policy_name	nvarchar(64)	varchar(64)	Scan policy associated with this scan
agent_name	nvarchar(256)	text	
scan_id	integer	integer	Scan ID
scan_type	integer	integer	0 = None 1 = File System Data 2 = Permissions 4 = Volume Free Space
triggered_start_time	datetime2(3)	timestamp	Initial time scan delegation starts
scan_start_time	datetime2(3)	timestamp	Start time when agent begins physical scan
scan_stop_time	datetime2(3)	timestamp	Stop time when agent completes physical scan
enum_start_time	datetime2(3)	timestamp	Agent metrics related to file system object enumeration

Column Name	SQL Server	PostgreSQL	Notes
enum_stop_time	datetime2(3)	timestamp	Agent metrics related to file system object enumeration
enum_file_count	integer	integer	Agent metrics related to file system object enumeration
enum_directory_count	integer	integer	Agent metrics related to file system object enumeration
enum_link_count	integer	integer	Agent metrics related to file system object enumeration
caching_start_time	datetime2(3)	timestamp	Metrics related to agent caching
caching_stop_time	datetime2(3)	timestamp	Metrics related to agent caching
cached_file_count	integer	integer	Metrics related to agent caching
cached_directory_count	integer	integer	Metrics related to agent caching
cached_link_count	integer	integer	Metrics related to agent caching
cache_size	integer	integer	Metrics related to agent caching
cache_size_max	integer	integer	Metrics related to agent caching
metadata_start_time	datetime2(3)	timestamp	Agent metrics related to filesystem metadata collection
metadata_stop_time	datetime2(3)	timestamp	Agent metrics related to filesystem metadata collection

Column Name	SQL Server	PostgreSQL	Notes
metadata_file_count	integer	integer	Agent metrics related to filesystem metadata collection
metadata_directory_count	integer	integer	Agent metrics related to filesystem metadata collection
metadata_link_count	integer	integer	Agent metrics related to filesystem metadata collection
accounts_start_time	datetime2(3)	timestamp	Agent metrics related to security principal collection
accounts_stop_time	datetime2(3)	timestamp	Agent metrics related to security principal collection
accounts_object_count	integer	integer	Agent metrics related to security principal collection
transfer_start_time	datetime2(3)	timestamp	Related to transfer of scan file from the Agent to the Engine
transfer_stop_time	datetime2(3)	timestamp	Related to transfer of scan file from the Agent to the Engine
db_start_time	datetime2(3)	timestamp	Database insert start time
db_stop_time	datetime2(3)	timestamp	Database insert stop time
status_code	integer	integer	Internal status code
error_string	nvarchar(1024)	varchar(1024)	

srs.scan_targets

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
network_path	nvarchar(256)	varchar(256)	Root path for scan target
network_path_lower	nvarchar(256)	[Not applicable]	Computed column
server	nvarchar(256)	varchar(256)	
identity_system_id	integer	integer	Reference to identity_systems table
platform	smallint	smallint	0 = Unknown 1 = Windows
filesystem	smallint	smallint	0 = Unknown 1 = NTFS
cost_per_unit	money	money	Not currently used

srs.scans

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
scan_policy_id	integer	integer	Reference to scan_policies table
triggered_start_time	datetime2(3)	timestamp	Initial time scan delegation starts
scan_start_time	datetime2(3)	timestamp	Start time when agent begins physical scan
scan_stop_time	datetime2(3)	timestamp	Stop time when agent completes physical scan
enum_start_time	datetime2(3)	timestamp	Agent metrics related to file system object enumeration
enum_stop_time	datetime2(3)	timestamp	Agent metrics related to file system object enumeration
enum_file_count	integer	integer	Agent metrics related to file system object enumeration
enum_directory_count	integer	integer	Agent metrics related to file system object enumeration
enum_link_count	integer	integer	Agent metrics related to file system object enumeration
caching_start_time	datetime2(3)	timestamp	Metrics related to agent caching
caching_stop_time	datetime2(3)	timestamp	Metrics related to agent caching
cached_file_count	integer	integer	Metrics related to agent caching
cached_	integer	integer	Metrics related to agent

Column Name	SQL Server	PostgreSQL	Notes
directory_count			caching
cached_link_count	integer	integer	Metrics related to agent caching
cache_size	integer	integer	Metrics related to agent caching
cache_size_max	integer	integer	Metrics related to agent caching
metadata_start_time	datetime2(3)	timestamp	Agent metrics related to filesystem metadata collection
metadata_stop_time	datetime2(3)	timestamp	Agent metrics related to filesystem metadata collection
metadata_file_count	integer	integer	Agent metrics related to filesystem metadata collection
metadata_directory_count	integer	integer	Agent metrics related to filesystem metadata collection
metadata_link_count	integer	integer	Agent metrics related to filesystem metadata collection
accounts_start_time	datetime2(3)	timestamp	Agent metrics related to security principal collection
accounts_stop_time	datetime2(3)	timestamp	Agent metrics related to security principal collection
accounts_object_count	integer	integer	Agent metrics related to security principal collection

Column Name	SQL Server	PostgreSQL	Notes
transfer_start_time	datetime2(3)	timestamp	Related to transfer of scan file from the Agent to the Engine
transfer_stop_time	datetime2(3)	timestamp	Related to transfer of scan file from the Agent to the Engine
db_start_time	datetime2(3)	timestamp	Database insert start time*
db_stop_time	datetime2(3)		Database insert stop time*
scan_type	integer	integer	0 = None 1 = File System Data 2 = Permissions 4 = Volume Free Space
scan_target_id	integer	integer	Reference to scan_targets table
local_identity_system_id	integer	integer	
retry_count	integer	integer	Current number of scan attempts
status_code	integer	integer	Internal status code
error_string	nvarchar(1024)	varchar(1024)	
progress_status	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress

Column Name	SQL Server	PostgreSQL	Notes
			3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending 6 = Previous 7 = Retained
next_retry_time	datetime2(0)	timestamp	Next scheduled time to retry a failed scan
ntfs_abe_enabled	bit	boolean	Flag indicating that the Windows share has ABE enabled
is_valid	bit	boolean	[Deprecated]
agent_name	nvarchar(256)	varchar(256)	

srs.security_descriptors

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
scan_data_id	bigint	bigint	Reference to scan_data table
control	integer	integer	Security descriptor control flags See https://docs.microsoft.com/en-us/windows/win32/secauthz/security-descriptor-control Possible flags: 0x0001 - Owner defaulted 0x0002 - Group defaulted 0x0004 - DACL present 0x0008 - DACL defaulted 0x0010 - SACL present 0x0020 - SACL defaulted 0x0100 - DACL auto inherit required 0x0200 - SACL auto inherit required 0x0400 - DACL auto Inherited 0x0800 - SACL auto inherited 0x1000 - DACL Protected (inheritance disabled) 0x2000 - SACL protected (inheritance disabled) 0x4000 - Resource Manager control is valid 0x8000 - Security Descriptor is self relative
dacl_present	bit	boolean	Indicates presence of DACL entries

Column Name	SQL Server	PostgreSQL	Notes
			for this security descriptor
sacl_present	bit	boolean	Indicates presence of SACL entries for this security descriptor

srs.tend_volume_freespace

Column Name	SQL Server	PostgreSQL	Notes
id	integer	integer	Primary key
scan_id	integer	integer	Scan ID
identity_system	nvarchar(256)	text	
network_path	nvarchar(max)	text	Scan target path
server	nvarchar(256)	text	
filesystem	integer	integer	0 = Unknown 1 = NTFS
volume_guid	uniqueidentifier	uuid	
volume_label	nvarchar(256)	text	
volume_bytes_total	bigint	bigint	
volume_bytes_free	bigint	bigint	
volume_bytes_used	bigint	bigint	
allocation_unit_size	integer	integer	
allocation_units_total	bigint	bigint	
allocation_units_free	bigint	bigint	
allocation_units_used	bigint	bigint	
status	integer	integer	
scan_time	datetime2(0)	timestamp	

5.2 - Temp Tables

tmp_cq_fs_paths

Column Name	SQL Server	PostgreSQL	Notes
report_id	integer	integer	Reference to primary key of associated srs.report_ definitions entry
scan_id	integer	integer	Reference to primary key of associated srs.scans entry
scan_type	integer	integer	0 = None 1 = File System Data 2 = Permissions 4 = Volume Free Space
progress_status	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending 6 = Previous 7 = Retained
scan_start_time	datetime3(2)	timestamp	Start time when agent begins physical scan

Column Name	SQL Server	PostgreSQL	Notes
scan_target_id	integer	integer	Reference to primary key of associated srs.scan_targets entry
target_path	nvarchar(max)	text	Selected path for this report
target_path_hash	binary(20)	bytea	SHA-1 hash of normalized target path
path_index	integer	integer	Used for hierarchical relation processing
ns_left	integer	integer	Nested-set Left index - used for hierarchical relation processing
ns_right	integer	integer	Nested-set Right index - used for hierarchical relation processing
path_depth	integer	integer	Entry depth with respect to the scan target's root path
path_type	integer	integer	0 = Unknown 1 = File 2 = Directory 3 = File Symbolic Link 4 = Directory Symbolic Link 5 = Junction 6 = Mount Point 7 = Share 8 = Volume 9 = DFS Link 10 = DFS Folder 11 = DFS Root

Column Name	SQL Server	PostgreSQL	Notes
			12 = HSM Stub 13 = Reparse Point Unknown 17 = Single Instance Storage Stub 18 = Named Stream
is_permission_scan	bit	boolean	Flag indicating whether this entry is for a file system permissions scan Can be used in place of scan_type
is_filesystem_scan	bit	boolean	Flag indicating whether this entry is for a file system metadata scan Can be used in place of scan_type
is_current	bit	boolean	Flag indicating whether this entry is for the current scan Can be used in place of progress_status
is_previous	bit	boolean	Flag indicating whether this entry is for a previous scan Can be used in place of progress_status
is_baseline	bit	boolean	Flag indicating whether this entry is for a baseline scan Can be used in place of progress_status

5.3 - Views

ad.ds_objects_view

Column Name	SQL Server	PostgreSQL	Notes
forest_dns	nvarchar (256)	varchar (256)	Forest DNS name
domain_dns	nvarchar (256)	varchar(256)	Account domain DNS name
domain_netbios	nvarchar(15)	varchar(15)	Account domain NetBIOS name
id	bigint	bigint	Primary key
dn	nvarchar (max)	text	Distinguished name
db_domain_sid	nvarchar (256)	varchar(256)	SID of the source domain where the object was scanned
domain_sid	nvarchar (256)	varchar(256)	SID of the account domain
db_last_update	datetime2(3)	timestamp	Last update time for this entry in the database
account_expires	datetime2(0)	timestamp	
create_timestamp	datetime2(0)	timestamp	
department	nvarchar(64)	varchar(64)	
description	nvarchar (1024)	varchar (1024)	Only uses first value of this multi-value attribute
display_name	nvarchar (256)	varchar(256)	
dns_host_name	nvarchar (2048)	varchar (2048)	Applies to Computer objects

Column Name	SQL Server	PostgreSQL	Notes
given_name	nvarchar(64)	varchar(64)	
group_type	integer	integer	See https://docs.microsoft.com/en-us/windows/win32/adschema/a-grouptype for details. Flags: 0x01 - System created group 0x02 - Global group 0x04 - Domain Local group 0x08 - Universal group 0x10 - APP_BASIC group for Windows Server Authorization Manager 0x20 - APP_QUERY group for Windows Server Authorization Manager 0x80000000 - Security Group. If not set, then a Distribution Group
last_logon_timestamp	datetime2(0)	timestamp	NOTE: This attribute only has 14-day granularity. See: https://docs.microsoft.com/en-us/windows/win32/adschema/a-lastlogontimestamp
mail	nvarchar (256)	varchar(256)	
managed_by_guid	nvarchar(36)	varchar(36)	GUID of referenced DS object
manager_guid	nvarchar(36)	varchar(36)	GUID of referenced DS object
object_category	nvarchar	varchar(256)	Using LDAP display name, not

Column Name	SQL Server	PostgreSQL	Notes
	(256)		FDN.
object_class	nvarchar (256)	varchar(256)	Only includes structural class value from this multi-value attribute.
object_guid	nvarchar(36)	varchar(36)	Object's GUID
object_sid	nvarchar (256)	varchar(256)	Object's Security Identifier
primary_group_sid	varbinary (68)	varchar(256)	SID of referenced object
sam_account_name	nvarchar (256)	varchar(256)	SAM account name
sam_account_type	integer	integer	See https://docs.microsoft.com/en-us/windows/win32/adschema/a-samaccounttype for details. Enum values: 0x00000000 - Domain 0x10000000 - Group 0x10000001 - Non-security Group object 0x20000000 - Alias object 0x20000001 - Non-security Alias object 0x30000000 - Normal User account 0x30000001 - Machine (computer) account 0x30000002 - Trust account

Column Name	SQL Server	PostgreSQL	Notes
			0x40000000 - APP_BASIC Group 0x40000001 - APP_QUERY Group
sam_principal_name	nvarchar (256)	varchar(256)	NetBIOS\SamAccountName. From msDS-PrincipalName. Note that the NetBIOS name here may be different from the associated domain NetBIOS name where this account was scanned. This is especially true for domain Builtin* accounts and foreign security principals.
surname	nvarchar(64)	varchar(64)	
title	nvarchar (128)	varchar(128)	
uac_flags	integer	integer	Combines both userAccessControl and msDs-User-Account-Control-Computed attribute values into a single flag. See the following for details: • https://docs.microsoft.com/en-us/windows/win32/adschema/a-useraccountcontrol • https://docs.microsoft.com/en-us/windows/win32/adschema/a-msds-user-account-control-computed Flags values: 0x00000001 - Logon script is

Column Name	SQL Server	PostgreSQL	Notes
			executed 0x00000002 - User Account disabled 0x00000008 - Home directory required 0x00000010 - Account currently locked out 0x00000020 - No password required 0x00000040 - User cannot change password 0x00000080 - User can send encrypted password 0x00000100 - Temporary duplicate account 0x00000200 - Normal account - typical user 0x00000800 - Inter-domain trust account 0x00001000 - Computer (Workstation / Member Server) account 0x00002000 - Domain controller computer account 0x00010000 - Password does not expire 0x00020000 - Majority Node Set (MNS) logon account 0x00040000 - Smart card required for logon 0x00080000 - Service account trusted for Kerberos delegation 0x00100000 - Account not allowed trust for delegation

Column Name	SQL Server	PostgreSQL	Notes
			0x00200000 - Account can only use DES keys 0x00400000 - Account does not require Kerberos pre-authentication for logon 0x00800000 - User password has expired 0x01000000 - Account enabled for delegation 0x04000000 - Partial secrets account 0x08000000 - Account can only use Use AES keys
upn	nvarchar (1024)	varchar (1024)	User principal name

srs.baseline_fs_scandata

Column Name	SQL Server	PostgreSQL	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
fullpath	nvarchar(max)	text	Full UNC path to the file system entry
name	nvarchar(256)	varchar(256)	File or directory name
filename_ extension	nvarchar(32)	varchar(32)	File name extension
create_time	datetime2(0)	timestamp	Stored as UTC time
modify_time	datetime2(0)	timestamp	Stored as UTC time
access_time	datetime2(0)	timestamp	Stored as UTC time
size	bigint	bigint	For files, actual size; for directories, accumulative size of all subordinate files
size_on_disk	bigint	bigint	Assumes typical allocation unit size of 4K
size_compressed	bigint	bigint	Only accurate for NTFS file systems
owner_identity_ system	nvarchar(256)	varchar(256)	Owner's Identity System name
owner_domain	nvarchar(256)	varchar(256)	Owner's Active Directory domain

Column Name	SQL Server	PostgreSQL	Notes
owner_name	nvarchar(256)	varchar(256)	SAM Account name
owner_fdn	nvarchar(512)	varchar(512)	Full distinguished object name
owner_display_name	nvarchar(max)	text	Domain\SamAccountName
owner_id	varbinary(68)	bytea	Security Identifier (SID)
attributes	integer	integer	0x0 = None 0x1 = Read Only 0x2 = Archive 0x4 = System 0x8 = Hidden 0x10 = Directory 0x20 = Compressed 0x40 = Offline 0x80 = NTFS device 0x100 = NTFS Normal 0x200 = NTFS Temporary 0x400 = NTFS Sparse File 0x800 = NTFS Reparse Point 0x1000 = NTFS Not content indexed 0x2000 = NTFS Encrypted 0x4000 = NTFS Virtual
attribute_string	nvarchar(256)	varchar(256)	See srs.attribute_string function
fullpath_hash	binary(20)	bytea	SHA-1 hash of lowercase fullpath

Column Name	SQL Server	PostgreSQL	Notes
idx	integer	integer	Scan index; unique per scan
parent_idx	integer	integer	Parent index. Used for hierarchical relation processing
path_depth	integer	integer	Entry depth with respect to the scan target's root path.
ns_left	integer	integer	Nested-set Left index – used for hierarchical relation processing
ns_right	integer	integer	Nested-set Right index – used for hierarchical relation processing
scan_id	integer	integer	Reference to scans table
scan_data_id	bigint	bigint	Reference to scan_data table
path_type	integer	integer	0 = Unknown 1 = File 2 = Directory 3 = File Symbolic Link 4 = Directory Symbolic Link 5 = Junction 6 = Mount Point 7 = Share 8 = Volume 9 = DFS Link 10 = DFS Folder 11 = DFS Root 12 = HSM Stub

Column Name	SQL Server	PostgreSQL	Notes
			13 = Reparse Point Unknown 17 = Single Instance Storage Stub 18 = Named Stream
status_code	integer	integer	

srs.baseline_fs_scans

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
scan_id	integer	integer	Reference to scans table
identity_system	nvarchar(256)	varchar(256)	Identity system name
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
platform	integer	integer	0 = Unknown 1 = Windows
filesystem	integer	integer	0 = Unknown 1 = NTFS
scan_type	integer	integer	Should always be 1
progress_status	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending

Column Name	SQL Server	PostgreSQL	Notes
			6 = Previous 7 = Retained
identity_system_id	integer	integer	
scan_target_id	integer	integer	
status_code	integer	integer	
ntfs_abe_enabled	bit	boolean	Flag indicating that the Windows share has ABE enabled
agent	nvarchar(256)	varchar(256)	Name of agent that performed the scan
file_count	integer	integer	Number of files in the scan
directory_count	integer	integer	Number of directories in the scan
link_count	integer	integer	Number of links (junctions, symbolic links, reparse points) in the scan

srs.baseline_ntfs_aces

Column Name	SQL Server	PostgreSQL	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
fullpath	nvarchar(max)	text	Full UNC path to the file system entry
trustee_identity_system	nvarchar(256)	varchar(256)	Trustee's Identity System name
trustee_domain	nvarchar(256)	varchar(256)	Trustee's Active Directory domain
trustee_name	nvarchar(256)	varchar(256)	SAMAccount name
trustee_fdn	nvarchar(512)	varchar(512)	Full distinguished name
trustee_display_name	nvarchar(max)	text	DOMAIN\SAMAccount
trustee_type	integer	integer	0 = Unknown / Other 1 = User 2 = Group 3 = Computer
sid	varbinary(68)	bytea	
access_mask	integer	integer	0x1 = Read Data / List Directory 0x2 = Write Data / Create File

Column Name	SQL Server	PostgreSQL	Notes
			0x4 = Append Data / Create Subdirectory 0x8 = Read Extended Attributes 0x10 = Write Extended Attributes 0x20 = File Execute / Traverse 0x40 = Delete Child 0x80 = Read Attributes 0x100 = Write Attributes 0x10000 = Delete 0x20000 = Read Permissions 0x40000 = Change Permissions 0x80000 = Change Owner 0x100000 = Synchronize 0x1000000 = Access System Security 0x10000000 = Generic All 0x20000000 = Generic Execute 0x40000000 = Generic Write 0x80000000 = Generic Read
access_mask_ string	nvarchar(128)	varchar(128)	See srs.access_mask_string
basic_ permissions	nvarchar(128)	varchar(128)	See srs.access_mask_ basic_string
ace_type	smallint	smallint	0 = Access Allowed

Column Name	SQL Server	PostgreSQL	Notes
			1 = Access Denied 2 = System Audit 9 = Allowed Callback 10 = Denied Callback 13 = System Audit Callback 17 = System Mandatory Label
ace_type_string	nvarchar(128)	varchar(128)	See srs.ace_type_string
ace_flags	smallint	smallint	0x1 = Object Inherit 0x2 = Container Inherit 0x4 = No Propagate 0x8 = Inherit Only 0x10 = Inherited 0x40 = Successful Access 0x80 = Failed Access
ace_flags_string	nvarchar(128)	varchar(128)	See srs.ace_flags_string
idx	integer	integer	Scan index; unique per scan
parent_idx	integer	integer	Parent index. Used for hierarchical relation processing
path_depth	integer	integer	Entry depth with respect to the scan target's root path.
ns_left	integer	integer	Nested-set Left index - used for hierarchical relation processing
ns_right	integer	integer	Nested-set Right index - used for hierarchical relation

Column Name	SQL Server	PostgreSQL	Notes
			processing
scan_id	integer	integer	Reference to scans table
scan_data_id	bigint	bigint	Reference to scan_data table
path_type	integer	integer	0 = Unknown 1 = File 2 = Directory 3 = File Symbolic Link 4 = Directory Symbolic Link 5 = Junction 6 = Mount Point 7 = Share 8 = Volume 9 = DFS Link 10 = DFS Folder 11 = DFS Root 12 = HSM Stub 13 = Reparse Point Unknown 17 = Single Instance Storage Stub 18 = Named Stream
status_code	integer	integer	
identity_system_id	integer	integer	Reference to identity_systems table
scan_target_id	integer	integer	Reference to scan_targets table

Column Name	SQL Server	PostgreSQL	Notes
ad_object_id	integer	integer	Reference to ad_objects table

srs.baseline_permissions_scans

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
scan_id	integer	integer	Reference to scans table
identity_system	nvarchar(256)	varchar(256)	Identity system name
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
platform	smallint	smallint	0 = Unknown 1 = Windows
filesystem	smallint	smallint	0 = Unknown 1 = NTFS
scan_type	integer	integer	Should always be 2
progress_status	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending

Column Name	SQL Server	PostgreSQL	Notes
			6 = Previous 7 = Retained
identity_system_id	integer	integer	Reference to identity_systems table
scan_target_id	integer	integer	Reference to scan_targets table
status_code	integer	integer	
ntfs_abe_enabled	bit	boolean	Flag indicating that the Windows share has ABE enabled
agent	nvarchar(256)	varchar(256)	Name of agent that performed the scan
directory_count	integer	integer	Number of directories in the scan

srs.current_fs_scandata

Column Name	SQL Server	PostgreSQL	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
fullpath	nvarchar(max)	text	Full UNC path to the file system entry
name	nvarchar(256)	varchar(256)	File or directory name
filename_ extension	nvarchar(32)	varchar(32)	File name extension
create_time	datetime2(0)	timestamp	Stored as UTC time
modify_time	datetime2(0)	timestamp	Stored as UTC time
access_time	datetime2(0)	timestamp	Stored as UTC time
size	bigint	bigint	For files, actual size; for directories, accumulative size of all subordinate files
size_on_disk	bigint	bigint	Assumes typical allocation unit size of 4K
size_compressed	bigint	bigint	Only accurate for NTFS file systems
owner_identity_ system	nvarchar(256)	varchar(256)	Owner's Identity System name
owner_domain	nvarchar(256)	varchar(256)	Owner's Active Directory domain

Column Name	SQL Server	PostgreSQL	Notes
owner_name	nvarchar(256)	varchar(256)	SAM Account name
owner_fdn	nvarchar(512)	varchar(512)	Full distinguished object name
owner_display_name	nvarchar(max)	text	Domain\SamAccountName
owner_id	varbinary(68)	bytea	Security Identifier (SID)
attributes	integer	integer	0x0 = None 0x1 = Read Only 0x2 = Archive 0x4 = System 0x8 = Hidden 0x10 = Directory 0x20 = Compressed 0x40 = Offline 0x80 = NTFS device 0x100 = NTFS Normal 0x200 = NTFS Temporary 0x400 = NTFS Sparse File 0x800 = NTFS Reparse Point 0x1000 = NTFS Not content indexed 0x2000 = NTFS Encrypted 0x4000 = NTFS Virtual
attribute_string	nvarchar(256)	varchar(256)	See srs.attribute_string function
fullpath_hash	binary(20)	bytea	SHA-1 hash of lowercase fullpath

Column Name	SQL Server	PostgreSQL	Notes
idx	integer	integer	Scan index; unique per scan
parent_idx	integer	integer	Parent index. Used for hierarchical relation processing
path_depth	integer	integer	Entry depth with respect to the scan target's root path.
ns_left	integer	integer	Nested-set Left index - used for hierarchical relation processing
ns_right	integer	integer	Nested-set Right index - used for hierarchical relation processing
scan_id	integer	integer	Reference to scans table
scan_data_id	bigint	bigint	Reference to scan_data table
path_type	integer	integer	0 = Unknown 1 = File 2 = Directory 3 = File Symbolic Link 4 = Directory Symbolic Link 5 = Junction 6 = Mount Point 7 = Share 8 = Volume 9 = DFS Link 10 = DFS Folder 11 = DFS Root 12 = HSM Stub

Column Name	SQL Server	PostgreSQL	Notes
			13 = Reparse Point Unknown 17 = Single Instance Storage Stub 18 = Named Stream
status_code	integer	integer	

srs.current_fs_scans

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
scan_id	integer	integer	Reference to scans table
identity_system	nvarchar(256)	varchar(256)	Identity system name
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
platform	integer	integer	0 = Unknown 1 = Windows
filesystem	integer	integer	0 = Unknown 1 = NTFS
scan_type	integer	integer	Should always be 1
progress_status	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending

Column Name	SQL Server	PostgreSQL	Notes
			6 = Previous 7 = Retained
identity_system_id	integer	integer	Reference to identity_systems table
scan_target_id	integer	integer	Reference to scan_targets table
status_code	integer	integer	
ntfs_abe_enabled	bit	boolean	Flag indicating that the Windows share has ABE enabled
is_valid	bit	boolean	[Deprecated]
agent	nvarchar(256)	varchar(256)	Name of agent that performed the scan
file_count	integer	integer	Number of files in the scan
directory_count	integer	integer	Number of directories in the scan
link_count	integer	integer	Number of links (junctions, symbolic links, reparse points) in the scan

srs.current_ntfs_aces

Column Name	SQL Server	PostgreSQL	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
fullpath	nvarchar(max)	text	Full UNC path to the file system entry
trustee_identity_system	nvarchar(256)	varchar(256)	Trustee's Identity System name
trustee_domain	nvarchar(256)	varchar(256)	Trustee's Active Directory domain
trustee_name	nvarchar(256)	varchar(256)	SAMAccount name
trustee_fdn	nvarchar(512)	varchar(512)	Full distinguished name
trustee_display_name	nvarchar(max)	text	DOMAIN\SAMAccount
trustee_type	integer	integer	0 = Unknown / Other 1 = User 2 = Group 3 = Computer
sid	varbinary(68)	bytea	
access_mask	integer	integer	0x1 = Read Data / List Directory 0x2 = Write Data / Create File

Column Name	SQL Server	PostgreSQL	Notes
			0x4 = Append Data / Create Subdirectory 0x8 = Read Extended Attributes 0x10 = Write Extended Attributes 0x20 = File Execute / Traverse 0x40 = Delete Child 0x80 = Read Attributes 0x100 = Write Attributes 0x10000 = Delete 0x20000 = Read Permissions 0x40000 = Change Permissions 0x80000 = Change Owner 0x100000 = Synchronize 0x1000000 = Access System Security 0x10000000 = Generic All 0x20000000 = Generic Execute 0x40000000 = Generic Write 0x80000000 = Generic Read
access_mask_ string	nvarchar(128)	varchar(128)	See srs.access_mask_string
basic_ permissions	nvarchar(128)	varchar(128)	See srs.access_mask_ basic_string
ace_type	smallint	smallint	0 = Access Allowed

Column Name	SQL Server	PostgreSQL	Notes
			1 = Access Denied 2 = System Audit 9 = Allowed Callback 10 = Denied Callback 13 = System Audit Callback 17 = System Mandatory Label
ace_type_string	nvarchar(128)	varchar(128)	See srs.ace_type_string
ace_flags	smallint	smallint	0x1 = Object Inherit 0x2 = Container Inherit 0x4 = No Propagate 0x8 = Inherit Only 0x10 = Inherited 0x40 = Successful Access 0x80 = Failed Access
ace_flags_string	nvarchar(128)	varchar(128)	See srs.ace_flags_string
idx	integer	integer	Scan index; unique per scan
parent_idx	integer	integer	Parent index. Used for hierarchical relation processing
path_depth	integer	integer	Entry depth with respect to the scan target's root path.
ns_left	integer	integer	Nested-set Left index - used for hierarchical relation processing
ns_right	integer	integer	Nested-set Right index - used for hierarchical relation

Column Name	SQL Server	PostgreSQL	Notes
			processing
scan_id	integer	integer	Reference to scans table
scan_data_id	bigint	bigint	Reference to scan_data table
path_type	integer	integer	0 = Unknown 1 = File 2 = Directory 3 = File Symbolic Link 4 = Directory Symbolic Link 5 = Junction 6 = Mount Point 7 = Share 8 = Volume 9 = DFS Link 10 = DFS Folder 11 = DFS Root 12 = HSM Stub 13 = Reparse Point Unknown 17 = Single Instance Storage Stub 18 = Named Stream
status_code	integer	integer	
identity_system_id	integer	integer	Reference to identity_systems table
scan_target_id	integer	integer	Reference to scan_targets table

Column Name	SQL Server	PostgreSQL	Notes
ad_object_id	integer	integer	Reference to ad_objects table

srs.current_permissions_scans

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
scan_id	integer	integer	Reference to scans table
identity_system	nvarchar(256)	varchar(256)	Identity system name
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
platform	smallint	smallint	0 = Unknown 1 = Windows
filesystem	smallint	smallint	0 = Unknown 1 = NTFS
scan_type	integer	integer	Should always be 2
progress_status	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending

Column Name	SQL Server	PostgreSQL	Notes
			6 = Previous 7 = Retained
identity_system_id	integer	integer	Reference to identity_systems table
scan_target_id	integer	integer	Reference to scan_targets table
status_code	integer	integer	
ntfs_abe_enabled	bit	boolean	Flag indicating that the Windows share has ABE enabled
is_valid	bit	boolean	[Deprecated]
agent	nvarchar(256)	varchar(256)	Name of agent that performed the scan
directory_count	integer	integer	Number of directories in the scan

srs.previous_fs_scandata

Column Name	SQL Server	PostgreSQL	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
fullpath	nvarchar(max)	text	Full UNC path to the file system entry
name	nvarchar(256)	varchar(256)	File or directory name
filename_ extension	nvarchar(32)	varchar(32)	File name extension
create_time	datetime2(0)	timestamp	Stored as UTC time
modify_time	datetime2(0)	timestamp	Stored as UTC time
access_time	datetime2(0)	timestamp	Stored as UTC time
size	bigint	bigint	For files, actual size; for directories, accumulative size of all subordinate files
size_on_disk	bigint	bigint	Assumes typical allocation unit size of 4K
size_compressed	bigint	bigint	Only accurate for NTFS file systems
owner_identity_ system	nvarchar(256)	varchar(256)	Owner's Identity System name
owner_domain	nvarchar(256)	varchar(256)	Owner's Active Directory domain

Column Name	SQL Server	PostgreSQL	Notes
owner_name	nvarchar(256)	varchar(256)	SAM Account name
owner_fdn	nvarchar(512)	varchar(512)	Full distinguished object name
owner_display_name	nvarchar(max)	text	Domain\SamAccountName
owner_id	varbinary(68)	bytea	Security Identifier (SID)
attributes	integer	integer	0x0 = None 0x1 = Read Only 0x2 = Archive 0x4 = System 0x8 = Hidden 0x10 = Directory 0x20 = Compressed 0x40 = Offline 0x80 = NTFS device 0x100 = NTFS Normal 0x200 = NTFS Temporary 0x400 = NTFS Sparse File 0x800 = NTFS Reparse Point 0x1000 = NTFS Not content indexed 0x2000 = NTFS Encrypted 0x4000 = NTFS Virtual
attribute_string	nvarchar(256)	varchar(256)	See srs.attribute_string function
fullpath_hash	binary(20)	bytea	SHA-1 hash of lowercase fullpath

Column Name	SQL Server	PostgreSQL	Notes
idx	integer	integer	Scan index; unique per scan
parent_idx	integer	integer	Parent index. Used for hierarchical relation processing
path_depth	integer	integer	Entry depth with respect to the scan target's root path.
ns_left	integer	integer	Nested-set Left index – used for hierarchical relation processing
ns_right	integer	integer	Nested-set Right index – used for hierarchical relation processing
scan_id	integer	integer	Reference to scans table
scan_data_id	bigint	bigint	Reference to scan_data table
path_type	integer	integer	0 = Unknown 1 = File 2 = Directory 3 = File Symbolic Link 4 = Directory Symbolic Link 5 = Junction 6 = Mount Point 7 = Share 8 = Volume 9 = DFS Link 10 = DFS Folder 11 = DFS Root 12 = HSM Stub

Column Name	SQL Server	PostgreSQL	Notes
			13 = Reparse Point Unknown 17 = Single Instance Storage Stub 18 = Named Stream
status_code	integer	integer	

srs.previous_fs_scans

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
scan_id	integer	integer	Reference to scans table
identity_system	nvarchar(256)	varchar(256)	Identity system name
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
platform	integer	integer	0 = Unknown 1 = Windows
filesystem	integer	integer	0 = Unknown 1 = NTFS
scan_type	integer	integer	Should always be 1
progress_status	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending

Column Name	SQL Server	PostgreSQL	Notes
			6 = Previous 7 = Retained
identity_system_id	integer	integer	
scan_target_id	integer	integer	
status_code	integer	integer	
ntfs_abe_enabled	bit	boolean	Flag indicating that the Windows share has ABE enabled
agent	nvarchar(256)	varchar(256)	Name of agent that performed the scan
file_count	integer	integer	Number of files in the scan
directory_count	integer	integer	Number of directories in the scan
link_count	integer	integer	Number of links (junctions, symbolic links, reparse points) in the scan

srs.previous_ntfs_aces

Column Name	SQL Server	PostgreSQL	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
fullpath	nvarchar(max)	text	Full UNC path to the file system entry
trustee_identity_system	nvarchar(256)	varchar(256)	Trustee's Identity System name
trustee_domain	nvarchar(256)	varchar(256)	Trustee's Active Directory domain
trustee_name	nvarchar(256)	varchar(256)	SAMAccount name
trustee_fdn	nvarchar(512)	varchar(512)	Full distinguished name
trustee_display_name	nvarchar(max)	text	DOMAIN\SAMAccount
trustee_type	integer	integer	0 = Unknown / Other 1 = User 2 = Group 3 = Computer
sid	varbinary(68)	bytea	
access_mask	integer	integer	0x1 = Read Data / List Directory 0x2 = Write Data / Create File

Column Name	SQL Server	PostgreSQL	Notes
			0x4 = Append Data / Create Subdirectory 0x8 = Read Extended Attributes 0x10 = Write Extended Attributes 0x20 = File Execute / Traverse 0x40 = Delete Child 0x80 = Read Attributes 0x100 = Write Attributes 0x10000 = Delete 0x20000 = Read Permissions 0x40000 = Change Permissions 0x80000 = Change Owner 0x100000 = Synchronize 0x1000000 = Access System Security 0x10000000 = Generic All 0x20000000 = Generic Execute 0x40000000 = Generic Write 0x80000000 = Generic Read
access_mask_ string	nvarchar(128)	varchar(128)	See srs.access_mask_string
basic_ permissions	nvarchar(128)	varchar(128)	See srs.access_mask_ basic_string
ace_type	smallint	smallint	0 = Access Allowed

Column Name	SQL Server	PostgreSQL	Notes
			1 = Access Denied 2 = System Audit 9 = Allowed Callback 10 = Denied Callback 13 = System Audit Callback 17 = System Mandatory Label
ace_type_string	nvarchar(128)	varchar(128)	See srs.ace_type_string
ace_flags	smallint	smallint	0x1 = Object Inherit 0x2 = Container Inherit 0x4 = No Propagate 0x8 = Inherit Only 0x10 = Inherited 0x40 = Successful Access 0x80 = Failed Access
ace_flags_string	nvarchar(128)	varchar(128)	See srs.ace_flags_string
idx	integer	integer	Scan index; unique per scan
parent_idx	integer	integer	Parent index. Used for hierarchical relation processing
path_depth	integer	integer	Entry depth with respect to the scan target's root path.
ns_left	integer	integer	Nested-set Left index - used for hierarchical relation processing
ns_right	integer	integer	Nested-set Right index - used for hierarchical relation

Column Name	SQL Server	PostgreSQL	Notes
			processing
scan_id	integer	integer	Reference to scans table
scan_data_id	bigint	bigint	Reference to scan_data table
path_type	integer	integer	0 = Unknown 1 = File 2 = Directory 3 = File Symbolic Link 4 = Directory Symbolic Link 5 = Junction 6 = Mount Point 7 = Share 8 = Volume 9 = DFS Link 10 = DFS Folder 11 = DFS Root 12 = HSM Stub 13 = Reparse Point Unknown 17 = Single Instance Storage Stub 18 = Named Stream
status_code	integer	integer	
identity_system_id	integer	integer	Reference to identity_systems table
scan_target_id	integer	integer	Reference to scan_targets table

Column Name	SQL Server	PostgreSQL	Notes
ad_object_id	integer	integer	Reference to ad_objects table

srs.previous_permissions_scans

Column Name	SQL Server	PostgreSQL	Notes
id	bigint	bigint	Primary key
scan_id	integer	integer	Reference to scans table
identity_system	nvarchar(256)	varchar(256)	Identity system name
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
platform	smallint	smallint	0 = Unknown 1 = Windows
filesystem	smallint	smallint	0 = Unknown 1 = NTFS
scan_type	integer	integer	Should always be 2
progress_status	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending

Column Name	SQL Server	PostgreSQL	Notes
			6 = Previous 7 = Retained
identity_system_id	integer	integer	Reference to identity_systems table
scan_target_id	integer	integer	Reference to scan_targets table
status_code	integer	integer	
ntfs_abe_enabled	bit	boolean	Flag indicating that the Windows share has ABE enabled
agent	nvarchar(256)	varchar(256)	Name of agent that performed the scan
directory_count	integer	integer	Number of directories in the scan

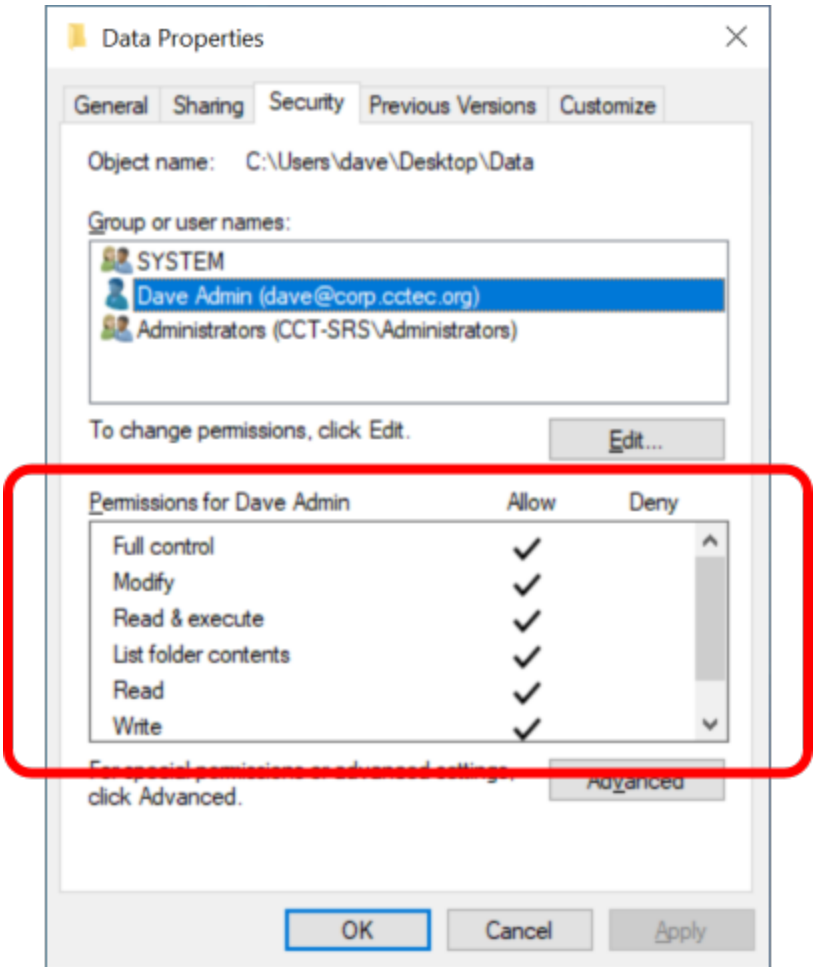
5.4 - Functions

srs.access_mask_basic_string

Parameters	SQL Server	PostgreSQL
@mask	integer	integer
@path_type	integer	integer
Return Value	nvarchar(128)	varchar(128)

Description: Converts an NTFS access mask value to its basic permissions string equivalent.

Note that the values displayed here are functionally equivalent to what is seen in the primary window of the security tab for an NTFS file system entry:



- Entries having permissions that do not fit the basic permissions (such as Special permissions) include an asterisk *.
- The path_type is required since the same flags represent different semantic values for folders, files and shares. Path type must be one of 1 (file), 2 (folder) or 7 (share)
- Permissions flags are mapped to one or more of the following values:
    - Full Control
    - Modify
    - Read & Execute
    - List Folder Contents (Folders only)
    - Read
    - Write
    - Special Permissions

Example (SQL Server)

```
SELECT TOP(100)
  sd.fullpath,
  srs.access_mask_basic_string(ntfs.access_mask, 2) AS basic_permissions
FROM srs.ntfs_aces AS ntfs
JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id
WHERE sd.path_type = 2;
```

Example (PostgreSQL)

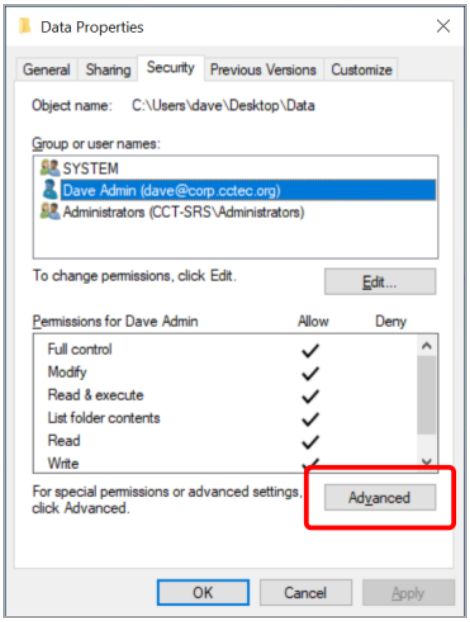
```
SELECT
  sd.fullpath,
  srs.access_mask_basic_string(ntfs.access_mask, 2) AS basic_permissions
FROM srs.ntfs_aces AS ntfs
JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id
WHERE sd.path_type = 2
LIMIT 100;
```

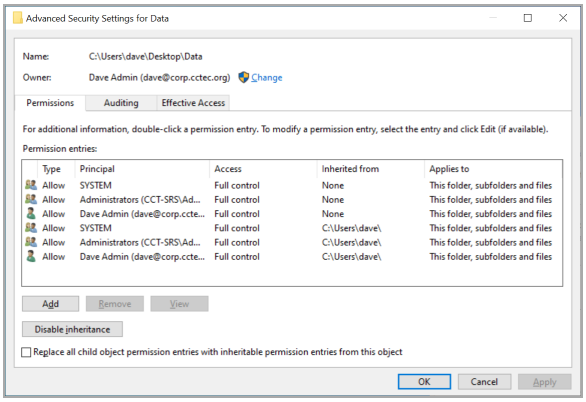
srs.access_mask_string

Parameters	SQL Server	PostgreSQL
@mask	integer	integer
@path_type	integer	integer
Return Value	nvarchar(128)	varchar(128)

Description: Converts an NTFS access mask value to its advanced permissions string equivalent.

Note that the values displayed here are functionally equivalent to what is seen in the advanced section of the security tab for an NTFS file system entry:





- The path_type is required since the same flags represent different semantic values for folders, files and shares. Path type must be one of 1 (file), 2 (folder) or 7 (share)
- Flags correspond to the following values:

0x00000001	RD	Read data / List folder
0x00000002	WD	Write data / Create file
0x00000004	AD	Append data / Create subdirectory
0x00000008	REA	Read extended attributes
0x00000010	WEA	Write extended attributes

0x00000020	X	File execute / Traverse
0x00000040	DC	Delete child (subdirectory)
0x00000080	RA	Read attributes
0x00000100	WA	Write attributes
0x00010000	D	Delete
0x00020000	RC	Read permissions
0x00040000	WDAC	Change permissions
0x00080000	WO	Change owner (take ownership)
0x00100000	S	Synchronize
0x01000000	AS	Access system security
0x10000000	GA	Generic All
0x20000000	GE	Generic Execute
0x40000000	GW	Generic Write
0x80000000	GR	Generic Read

Example (SQL Server)

```
SELECT TOP(100)
  sd.fullpath,
  srs.access_mask_string(ntfs.access_mask, sd.path_type) AS access_mask
FROM srs.ntfs_aces AS ntfs
JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id;
```

Example (PostgreSQL)

```
SELECT
  sd.fullpath,
  srs.access_mask_string(ntfs.access_mask, sd.path_type) AS access_mask
FROM srs.ntfs_aces AS ntfs
JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id
LIMIT 100;
```

srs.ace_flags_string

Parameters	SQL Server	PostgreSQL
@flags	integer	integer
Return Value	nvarchar(128)	varchar(128)

Description: Converts the access mask flags to a string representation. Flags are converted as follows:

0x001	(OI)	Object inherit
0x002	(CI)	Container inherit
0x004	(NP)	No propagate
0x008	(IO)	Inherit only
0x010	(I)	Inherited
0x040	(SA)	Successful access
0x080	(FA)	Failed access

Example (SQL Server)

```
1  SELECT TOP(100)
2     sd.fullpath,
3     srs.access_mask_string(ntfs.access_mask, sd.path_type) AS access_mask,
4     srs.ace_flags_string(ntfs.flags) AS ace_flags
5  FROM srs.ntfs_aces AS ntfs
6  JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id;
```

Example (PostgreSQL)

```
1  SELECT
2     sd.fullpath,
3     srs.access_mask_string(ntfs.access_mask, sd.path_type) AS access_mask,
4     srs.ace_flags_string(ntfs.flags) AS ace_flags
5  FROM srs.ntfs_aces AS ntfs
6  JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id
7  LIMIT 100;
```

srs.ace_type_string

Parameters	SQL Server	PostgreSQL
@ace_type	integer	integer
Return Value	nvarchar(128)	varchar(128)

Description: Converts the access mask type value to a corresponding text value.

Flags correspond as follows:

0	Access Allowed
1	Access Denied
2	System Audit
3	System Alarm
4	Allowed Compound
5	Allowed Object
6	Denied Object
7	System Audit Object
8	System Alarm Object
9	Allowed Callback
10	Denied Callback
11	Allowed Callback Object
12	Denied Callback Object
13	System Audit Callback
14	System Alarm Callback
15	System Audit Callback Object
16	System Alarm Callback Object
17	System Mandatory Label

For NTFS file systems the primary values of concern are Allowed (0), Denied (1), Audit (2), and System Mandatory Label (17).

Example (SQL Server)

```
SELECT TOP(100)
  sd.fullpath,
  srs.access_mask_string(ntfs.access_mask, sd.path_type) AS access_mask,
  srs.ace_flags_string(ntfs.flags) AS ace_flags,
  srs.ace_type_string(ntfs.ace_type) AS ace_type
```

```
FROM srs.ntfs_aces AS ntfs
JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id;
```

Example (PostgreSQL)

```
SELECT sd.fullpath,
   srs.access_mask_string(ntfs.access_mask, sd.path_type) AS access_mask,
   srs.ace_flags_string(ntfs.flags) AS ace_flags,
   srs.ace_type_string(ntfs.ace_type) AS ace_type
FROM srs.ntfs_aces AS ntfs
JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id
LIMIT 100;
```

srs.ad_account_name

Parameters	SQL Server	PostgreSQL
@domain	nvarchar(1024)	varchar(1024)
@name	nvarchar(1024)	varchar(1024)
@sid	binary(68)	bytea
Return Value	nvarchar(max)	text

Description: Converts primary naming values for an Windows security principal to a display name.

- If domain is null or empty, the leading backslash is not included in the result
- If the name is null or empty, the result value is the SDDL sid representation
- If the sid is needed but is invalid, the return value is [Invalid SID]

Example - Domain and Name

```
1 | SELECT srs.ad_account_name('BUILTIN', 'Administrators', null);
```

Example - SID

```
1
2 | SELECT srs.ad_account_name('', '', 0x0102000000000005200000020020000);
```

srs.attribute_string

Parameters	SQL Server	PostgreSQL
@flags	integer	integer
Return Value	nvarchar(256)	varchar(256)

Description: Converts an attributes value to its equivalent string representation. Flags correspond to the following values:

0x00000000		None
0x00000001	Ro	Read Only
0x00000002	Ar	Archive
0x00000004	Sy	System
0x00000008	Hi	Hidden
0x00000010	Dr	Directory
0x00000020	Co	Compressed
0x00000040	Ol	Offline
0x00000080	De	NTFS device
0x00000100	No	NTFS Normal
0x00000200	Te	NTFS Temporary
0x00000400	Sp	NTFS Sparse File
0x00000800	Rp	NTFS Reparse Point
0x00001000	Nc	NTFS Not content indexed
0x00002000	En	NTFS Encrypted
0x00004000	Vi	NTFS Virtual

Example (SQL Server)

```
1  SELECT TOP(100)
2    fullpath,
3    srs.attribute_string(attributes)
4  FROM srs.scan_data;
```

Example (PostgreSQL)

```
1  SELECT
2    fullpath,
```

```
3     srs.attribute_string(attributes)
4   FROM srs.scan_data
5   LIMIT 100;
```

srs.byte_string

Parameters	SQL Server	PostgreSQL
@size	bigint	bigint
Return Value	nvarchar(64)	text

Description: Converts a size value to a string representation of the closest unit.

- The return value has a maximum precision of two decimal places.
- Units include kilobyte (KB), megabyte (MB), gigabyte (GB), terabyte (TB), petabyte (PB), and exabyte (EB).

Example

```
1 | SELECT srs.byte_string(1287168);
```

srs.byte_unit_string

Parameters	SQL Server	PostgreSQL
@size	bigint	bigint
@unit	nvarchar(10)	varchar(10)
@precision	integer	integer
Return Value	nvarchar(64)	text

Description: Converts a number to a string representation of the specified unit with the specified precision.

- The specified precision is limited to a value from 0 to 3. Values outside this range will be adjusted to 0 or 3 accordingly.
- Unit specifiers are case insensitive and include:
  - byte
  - KB (kilobyte)
  - MB (megabyte)
  - GB (gigabyte)
  - TB (terabyte)
  - PB (petabyte)
  - EB (exabyte)

Example

```
1 | SELECT srs.byte_unit_string(1287168, 'KB', 3)
```

srs.bytes_to_hex_string

Parameters	SQL Server	PostgreSQL
@byte_sequence	varbinary(max)	bytea
Return Value	nvarchar(max)	text

Description: Converts a byte sequence to its equivalent hex string representation.

- Returned hex string is lower case with no separators and no prefix.

Example

```
1 SELECT
2    srs.bytes_to_hex_string(ad.sid)
3 FROM srs.ad_objects AS ad;
```

srs.guid_bytes

Parameters	SQL Server	PostgreSQL
@guid_text	nvarchar(38)	varchar(38)
Return Value	varbinary(16)	bytea

Description: Converts a compatible GUID text string to its equivalent binary representation.

Recommended input format: *xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx*

- Surrounding curly braces are optional
- Hex values A-F may be in upper or lower case
- Hyphen separators must be present at the specified 4 locations or not at all.

Example

```
1 | SELECT srs.guid_bytes('12345678-1234-5678-9abc-123456789abc');
```

srs.guid_text

Parameters	SQL Server	PostgreSQL
@guid_binary	varbinary(16)	bytea
Return Value	nvarchar(36)	varchar(36)

Description: Converts a binary GUID value to its equivalent string representation.

Note that returned strings are in the canonical lower-case GUID format *xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx*.

Example

```
1 | SELECT fdn, srs.guid_text(guid) FROM srs.ad_objects WHERE id=1;
```

srs.hex_string_to_bytes

Parameters	SQL Server	PostgreSQL
@byte_sequence	varbinary(max)	bytea
Return Value	nvarchar(max)	text

Description: Converts a hex string to its equivalent bytes.

- Hex values A-F may be in upper or lower case
- Hex string must be a proper string with an even number of characters – leading zeros are required for each hex value having a single digit.
- Do not include separators such as hyphens between hex values

Example

```
1 SELECT srs.hex_string_to_bytes('01ab3d4407');
```

srs.path_hash

Parameters	SQL Server	PostgreSQL
@path	nvarchar(max)	text
Return Value	binary(20)	bytea

Description: Returns the binary SHA-1 hash for a given path.

- The input path is first converted to lower case
- The input path is then converted to byte representation using the default text encoding of the database for string values (typically UTF-8 on PostgreSQL, and Unicode UCS-2 on SQL Server)
- Useful for finding a path in the srs.scan_data table using the fullpath_hash indexed column

Example

```
1 SELECT * FROM srs.scan_data
2 WHERE fullpath_hash = srs.path_hash('\\server-1.ad.cctec.org\Users\user1');
```

srs.path_hash_sha256

Parameters	SQL Server	PostgreSQL
@path	nvarchar(max)	text
Return Value	binary(32)	bytea

Description: Returns the binary SHA256 hash for a given path.

- The input path is first converted to lower case
- The input path is then converted to byte representation using the default text encoding of the database for string values (typically UTF-8 on PostgreSQL, and Unicode UCS-2 on SQL Server)
- Useful for finding a path (web URL) in the ms365.drive_items table using the web_url_hash indexed column

Example

```
SELECT * FROM ms365.drive_items
WHERE web_url_hash = srs.path_hash_sha256
('https://mysite.sharepoint.com/sites/Shared%20Documents');
```

srs.sid_bytes

Parameters	SQL Server	PostgreSQL
@sid	nvarchar(256)	varchar(256)
Return Value	varbinary(68)	bytea

Description: Converts an SDDL representation of a Security Identifier value to its binary form.

- Input SID values must be in proper SDDL form

Example

```
1 | SELECT * FROM srs.ad_objects WHERE srs.sid_bytes('S-1-5-32-544') = sid;
```

srs.sid_text

Parameters	SQL Server	PostgreSQL
@sid_bytes	varbinary(68)	bytea
Return Value	nvarchar(256)	varchar(256)

Description: Converts a binary Security Identifier to its SDDL string representation.

Example

```
1 | SELECT domain, name, srs.sid_text(sid) FROM srs.ad_objects;
```