Micro Focus File Reporter 4.0

Database Schema and Custom Queries Guide

February 4, 2021

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
122 North Laurens St.
Greenville, SC, 29601
U.S.A.
http://condrey.co

For information about Micro Focus legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, see https://www.novell.com/company/legal/.

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 in order to function. Further, these elements may require appropriate configuration and resources such as computing, memory, storage, or bandwidth in order 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 these elements must fully adhere to and support the appropriate APIs, standards, and protocols in order 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

This reference guide is written to provide database administrators comprehensive information for understanding and generating Custom Query reports through Micro Focus File Reporter.

Audience

This manual is intended for database administrators who want to generate Custom Query reports using File Reporter 4.0.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comment feature at the bottom of each page of the online documentation, or go to www.novell.com/documentation/feedback.html and enter your comments there.

Documentation Updates

For the most recent version of the *Micro Focus File Reporter 4.0 Database Schema and Custom Queries Guide*, visit the Micro Focus File Reporter Documentation Web site.

Additional Documentation

For additional Micro Focus File Reporter documentation, see the following guides at the Micro Focus File Reporter Documentation Web site:

- *Micro Focus File Reporter 4.0 Installation Guide*
- *Micro Focus File Reporter 4.0 Administration Guide*

Documentation Conventions

In this documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

When a single pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname is presented with a backslash. Users of platforms that require a forward slash, such as Linux*, should use forward slashes as required by your software.

When a startup switch can be written with a forward slash for some platforms or a double hyphen for other platforms, the startup switch is presented with a forward slash. Users of platforms that require a double hyphen, such as Linux, should use double hyphens as required by your software.

1 Overview

1.1 Updates

1.1.1 New Schema for Microsoft 365

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

1.1.2 Added Hex String Functions

The following supported functions have been added for use with conversion to and from bytes and their hex string equivalents:

- srs.bytes_to_hex_string
- srs.hex_string_to_bytes

1.1.3 Added Path Hash Function

The following function was added for support of sha256 hashing of path strings:

- srs.path_hash_sha256

It operates in the same way as the previous srs.path_hash function in that it converts the input string to lower case prior to calculating the hash which in this case is SHA-256.

Currently this function is used for web URL path hashes in Microsoft 365 entries.

1.2 Breaking Changes

1.2.1 Removed Support for Open Enterprise Server

Support for Open Enterprise Server file systems such as NSS and eDirectory identity systems have been removed for the 4.0 release. The following tables, views, and functions are no longer present:

- srs.edir_ds_trustees
- srs.edir_objects
- srs.edir_security_equals
- srs.ncp_trustees
- srs.current_fs_scandata_edir
- srs.current_ncp_trustees
- srs.previous_fs_scandata_edir
- srs.previous_ncp_trustees
- srs.baseline_fs_scandata_edir
- srs.baseline_ncp_trustees
- srs.ncp_rights_string

Support for eDirectory and Open Enterprise Server will continue with the previous 3.*x* release.

1.2.2 Deprecated Views

The following views are now deprecated 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

1.3 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 meta data, permissions

- File Content Analysis Data – data related to discovery of search expressions over file content
- Microsoft 365 Data – data related to drives, drive items and supporting meta data and permissions as well as basic teams and sites info in Microsoft 365

Although any tables, views, stored procedures and functions in the database can be accessed via custom queries, only the tables, views, and functions listed here are supported.

IMPORTANT: For users who are new to SQL, the supported views might be 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 latest Current scan data.

More experienced administrators however, will find that performance benefits can arise from making direct inline queries against the tables themselves, especially for complex scenarios.

1.4 Schema Namespace

All supported database objects and functions reside in specific schema namespaces. For example, the distinguished name for the table `scan_data` would be referenced as `srs.scan_data` when using the namespace prefix.

Although use of the namespace prefix is not required in all cases, there are some cases where it is required, such as when referencing a user defined function in Microsoft SQL Server, or when another database object of the same name exists in the schema search path. For these reasons you should always reference each supported database object and function with its documented namespace prefix.

1.5 Supported Tables

Table 1-1 *Supported Database 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 current 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

Category	Table Name	Notes
	srs.ntfs_aces	Scanned NTFS ACEs
	srs.security_descriptors	Scanned NTFS security descriptors
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_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.site_drives	List of discovered MS365 drives
	ms365.sites	List of discovered MS365 SharePoint sites
	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

1.6 Supported Views

Table 1-2 *Supported Database 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 Previous permissions scans
	srs.baseline_fs_scans	List of Baseline file system scans
	srs.baseline_permissions_scans	List of Baseline permissions scans
	srs.current_fs_scandata_ad	All file system scan data from Current scans in Active Directory environments
	srs.current_fs_scandata	Combined list of all Current file system scan data
	srs.previous_fs_scandata_ad	All file system scan data from Previous scans in Active Directory environments
	srs.previous_fs_scandata	Combined list of all Previous file system scan data
	srs.baseline_fs_scandata_ad	All file system scan data from Baseline scans in Active Directory environments
	srs.baseline_fs_scandata	Combined list of all Baseline file system scan data
	srs.current_ntfs_aces	All Current permissions scan data in Active Directory environments
	srs.previous_ntfs_aces	All Previous permissions scan data in Active Directory environments
	srs.baseline_ntfs_aces	All Baseline permissions scan data in Active Directory environments

1.7 Supported Functions

Table 1-3 *Supported Database Functions*

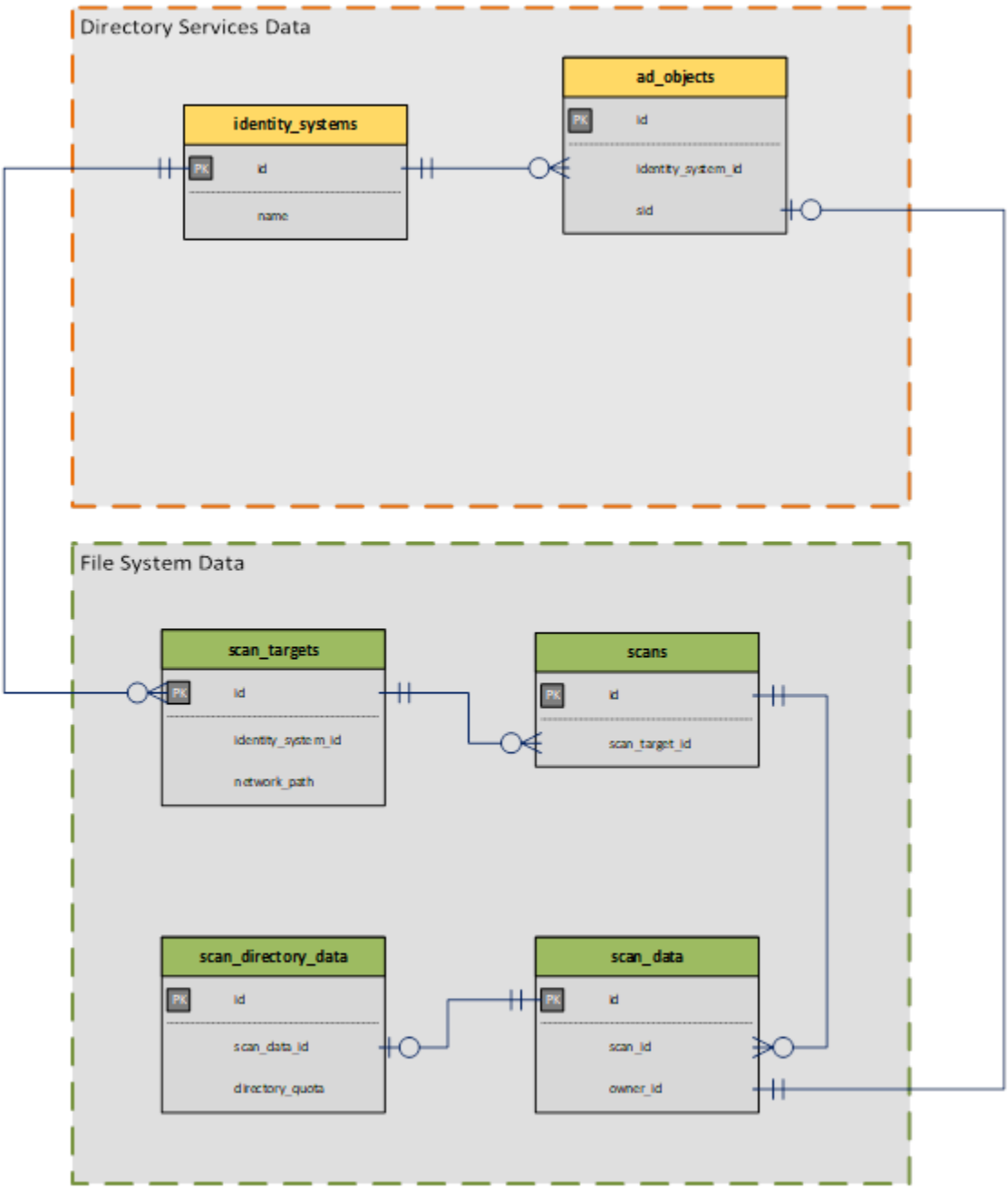
Category	View Name	Description
General	srs.byte_string	Converts raw number to byte string such as 10 MB or 3.25 KB
	srs.attribute_string	Converts attributes to string representation
	srs.guid_bytes	Converts Guid from string to binary

Category	View Name	Description
	srs.guid_text	Converts Guid from binary to string
	srs.path_hash	Calculates SHA-1 hash of full path
	srs.sid_bytes	Converts SID from string to binary
Identity Systems	srs.sid_text	Converts SID from binary to string
	srs.access_mask_basic_string	Converts access mask value to basic permissions string
Permissions	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
	srs.ad_account_name	Combines AD account name elements to a single display name

2 Custom Query

2.1 Understanding Table Relationships

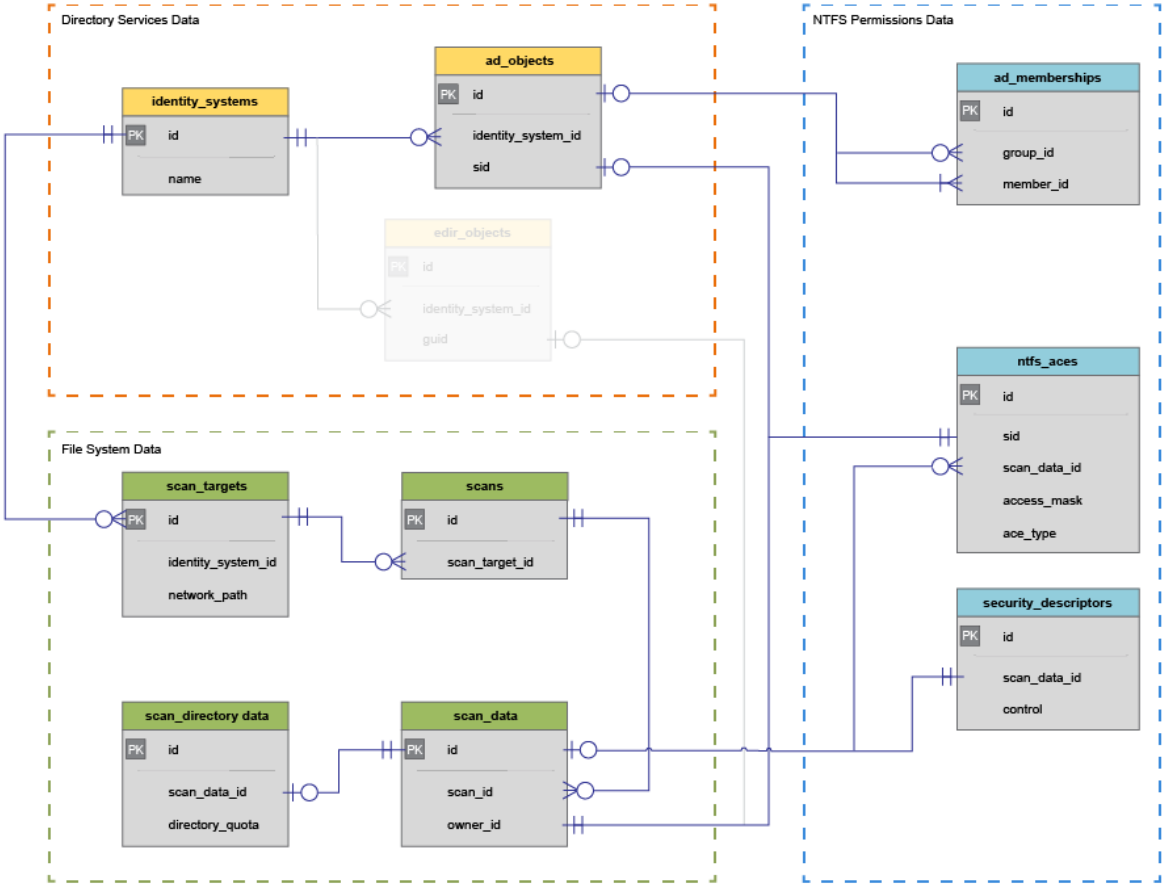
2.1.1 Windows File System Metadata



The collected scan data is generally broken down into three major areas: Identity System info, File System data, and Permissions data.

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

2.1.2 Windows File System Permissions



NTFS Permissions data is limited to folder structure as well as assigned and inherited NTFS access control entries (ACEs).

It should be noted that 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 currently collected.

2.2 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.

- Section 2.2.6, “Scope by Security Principal,” on page 20
- Section 2.2.7, “Basic Filtering,” on page 21

2.2.1 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.

Example: Select file system data from a given identity system, limited to 100 entries.

SQL Server

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

PostgreSQL

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

2.2.2 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.

Example: Select all file system data from a specific server, limited to 100 entries.

SQL Server

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

PostgreSQL

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

2.2.3 Scope by Scan Target

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

Note that the scan target name might 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.

SQL Server

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

PostgreSQL

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

2.2.4 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.	For all sibling file system entries, they will have the same parent index.
path_depth	Current path depth relative to 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 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 detail.

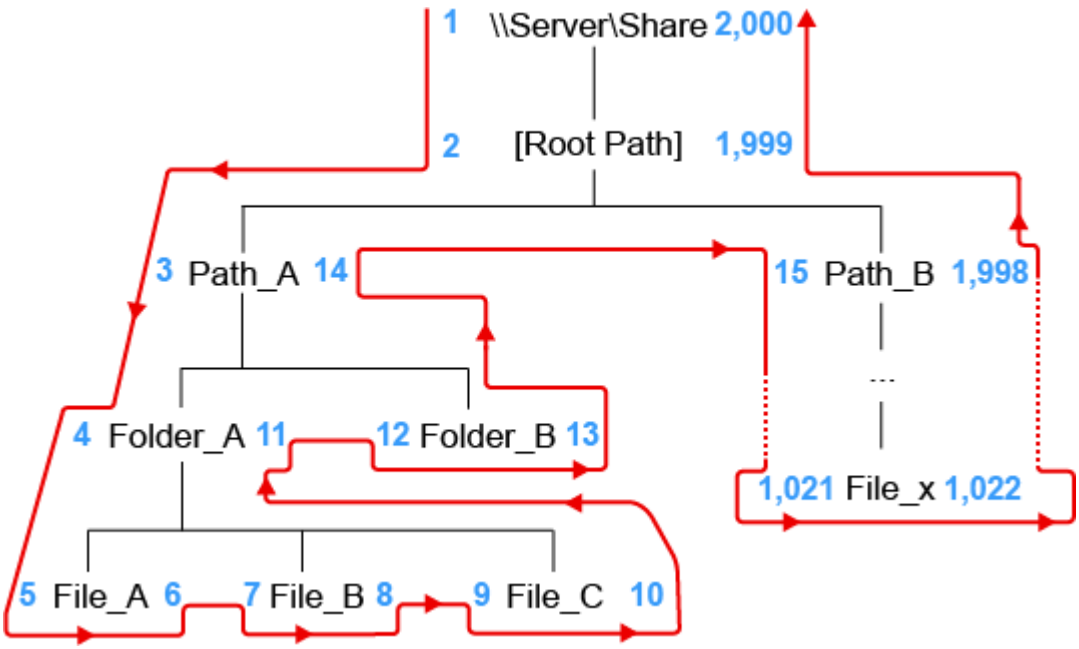
Example: Select all NTFS file system entries subordinate to, and including the specified target path.

```
WITH root_path AS (
    SELECT sd.fullpath, sd.ns_left, sd.ns_right, sd.path_type, sd.scan_id
    FROM srs.current_fs_scandata_ad AS sd
    WHERE sd.fullpath_hash = srs.path_hash('\\server1.ad.test.lab\Share\path\subpath')
    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;
```

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.

Figure 2-1 *Nested Set Calculations Example*



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 http://en.wikipedia.org/wiki/Nested_set_model.

2.2.5 Scope by Directory with Path Depth Limit

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.

Example: Select all paths starting two levels below a given path.

```
WITH root_path AS (
    SELECT sd.fullpath, sd.ns_left, sd.ns_right, sd.path_type, 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:

```
WITH root_path AS (
    SELECT sd.fullpath, sd.ns_left, sd.ns_right, sd.path_type, 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
```

2.2.6 Scope by Security Principal

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

Example: Select all files for a given server owned by a specific AD user, limited to 100 entries.

SQL Server

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

PostgreSQL

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

Example: Select all folders where a user is a direct trustee (not inherited) for NTFS, limited to 100 entries.

SQL Server

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

PostgreSQL

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

2.2.7 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" on page 21
- "Filter by File Extension" on page 21
- "Filter by Date Range" on page 22
- "Filter by File Name" on page 22

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.

Also 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 a 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: http://msdn.microsoft.com/en-us/library/ms190301

Postgres: http://www.postgresql.org/docs/current/static/functions-matching.html

3 Custom Schema Reference

3.1 Tables

3.1.1 ANALYSIS.FILE_SCAN_ENTRIES

Table 3-1 *File Scan Summary Entries Table Definition*

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
id	bigint	bigint	Primary key
scan_time	datetime2(3)	timestamp without time zone	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 without time zone	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
status_code	int	int	Processing status code for this file entry

3.1.2 MS365.DRIVE_ITEM_TYPES

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

3.1.3 MS365.DRIVE_ITEMS

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
id	bigint	bigint	Primary key
scan_id	bigint	bigint	Reference to primary key in ms365.drive_scans
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 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 associated identity
create_time	datetime2(3)	timestamp	Create time for entry
item_type	int	int	0 = unknown 1 = file 2 = folder 3 = remote item
file_hash	varbinary(64)	bytea	Files only - QuickXorHash of entry See: https://docs.microsoft.com/en-us/graph/api/resources/hashes?view=graph-rest-1.0

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
child_count	bigint	bigint	Folders only – number of child entries in the folder (only includes one level deep, not recursive)
modified_by	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for associated identity
modify_time	datetime2(3)	timestamp	Last modified time
name	nvarchar(256)	varchar(256)	Name of entry
file_extension	nvarchar(32)	varchar(32)	File name extension
size	bigint	bigint	Size in bytes
web_url	nvarchar(max)	text	Full path to item
web_url_hash	varbinary(32)	bytea	sha-256 hash of web_url

3.1.4 MS365.DRIVE_SCANS

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
id	bigint	bigint	Primary key
job_id	int	int	Reference to primary key in ms365.jobs
drive_id	bigint	bigint	Reference to primary key in ms365.drives
scan_status	int	int	0 = Queued 1 = In progress 2 = Completed 3 = Failed 99 = Canceled
scan_state	int	int	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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
agent_name	nvarchar(256)	varchar(256)	Name of Agent365 server performing the scan

3.1.5 MS365.DRIVE_SCANS_HISTORY

Column Name	SQL Server Data Type	PostgreSQL Data Type	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	int2 = Completed	0 = Queued 1 = In progress 2 = Completed 3 = Failed 99 = Canceled
scan_state	int	int	0 = Pending 1 = Current 99 = Marked for cleanup
result_string	nvarchar(max)	text	Success or error message

3.1.6 MS365.DRIVES

Column Name	SQL Server Data Type	PostgreSQL Data Type	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
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

3.1.7 MS365.GROUP_DRIVES

Column Name	SQL Server Data Type	PostgreSQL Data Type	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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
ms365_group_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for associated group
ms365_drive_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for associated drive

3.1.8 MS365.GROUP_MEMBER_TYPES

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

3.1.9 MS365.GROUP_MEMBERS

Column Name	SQL Server Data Type	PostgreSQL Data Type	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 associated group
ms365_member_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for associated member
member_type	int	int	0 = direct 1 = transitive

3.1.10 MS365.GROUP_SITES

Column Name	SQL Server Data Type	PostgreSQL Data Type	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 associated group
ms365_site_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for associated SharePoint site

3.1.11 MS365.GROUPS

Column Name	SQL Server Data Type	PostgreSQL Data Type	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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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

3.1.12 MS365.IDENTITY_TYPES

Column Name	SQL Server Data Type	PostgreSQL Data Type	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

3.1.13 MS365.JOBS

Column Name	SQL Server Data Type	PostgreSQL Data Type	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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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

3.1.14 MS365.JOBS_HISTORY

Column Name	SQL Server Data Type	PostgreSQL Data Type	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

3.1.15 MS365.PERMISSIONS

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
id	bigint	bigint	Primary key
scan_id	bigint	bigint	Reference to primary key in ms365.drive_scans
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_id_type	nvarchar(64)	varchar(64)	One of: ◆ user ◆ application ◆ device
grantedto_ms365_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for associated trustee
grantedto_display_name	nvarchar(256)	varchar(256)	Friendly name 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 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 associated application

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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: ◆ 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 ◆ sp.full control See: https://docs.microsoft.com/en-us/graph/api/resources/permission?view=graph-rest-1.0

3.1.16 MS365.SHARING_LINK_MEMBERS

Column Name	SQL Server Data Type	PostgreSQL Data Type	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
display_name	nvarchar(256)	varchar(256)	Friendly name of member

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
ms365_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for associated member

3.1.17 MS365.SITE_DRIVES

Column Name	SQL Server Data Type	PostgreSQL Data Type	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_site_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for associated SharePoint site
ms365_drive_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for associated drive

3.1.18 MS365.SITES

Column Name	SQL Server Data Type	PostgreSQL Data Type	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
ms365_parent_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for associated parent site
display_name	nvarchar(256)	varchar(256)	Friendly name of SharePoint site

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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

3.1.19 MS365.TEAM_CHANNELS

Column Name	SQL Server Data Type	PostgreSQL Data Type	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 associated path
ms365_files_folder_drive_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for associated path’s drive

3.1.20 MS365.TEAMS

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
id	bigint	bigint	Primary key
job_id	int	int	Reference to primary key in ms365.jobs

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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

3.1.21 MS365.TENANTS

Column Name	SQL Server Data Type	PostgreSQL Data Type	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

3.1.22 MS365.USER_DRIVES

Column Name	SQL Server Data Type	PostgreSQL Data Type	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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
ms365_user_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for associated user
ms365_drive_id	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for associated drive

3.1.23 MS365.USERS

Column Name	SQL Server Data Type	PostgreSQL Data Type	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_immutable_id	nvarchar(256)	varchar(256)	Unique id mapping synced on-prem user to associated MS365 user
account_enabled	bit	boolean	Account is enabled

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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

3.1.24 SRS.AD_MEMBERSHIPS

Table 3-2 *Active Directory Memberships Table Definition*

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
id	bigint	bigint	Primary key
group_id	integer	integer	
member_id	integer	integer	

3.1.25 SRS.AD_OBJECTS

Table 3-3 *Active Directory Objects Table Definition*

Column Name	SQL Server Data Type	PostgreSQL Data Type	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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
guid	binary(16)	bytea	Globally Unique Identifier
sid	varbinary(68)	bytea	Security Identifier
object_type	integer	integar	0 = Unknown / Other 1 = User 2 = Group 3 = Computer
identity_system_id	integer	integer	Reference to primary key of identity systems table

3.1.26 SRS.IDENTITY_SYSTEMS

Table 3-4 *Identity Systems Table Definition*

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
id	integer	integer	Primary key
type	integer	integer	0 = Unknown 1 = Active Directory 2 = eDirectory 3 = Windows Local
name	nvarchar(256)	varchar(256)	Identity system name eDirectory - Tree name Active Directory – Forest FDN
domain	nvarchar(256)	varchar(256)	Active Directory domain
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 without timezone	

3.1.27 SRS.NTFS_ACES

Table 3-5 *NTFS ACEs Table Definition*

Column Name	SQL Server Data Type	PostgreSQL Data Type	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 0x20 = File Execute / Traverse 0x40 = Delete Child 0x80 = Read Attributes 0x100 = Write Attributes 0x10000 = Delete 0x20000 = Read Permissions 0x40000 = Change Permissions0x80000 = Change Owner 0x100000 = Synchronize 0x1000000 = Access System Security 0x10000000 = Generic All 0x20000000 = Generic Execute 0x40000000 = Generic Write 0x80000000 = Generic Read

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
sid	varbinary(68)	bytea	Trustee SID
index_on_disk	smallint	smallint	Discovered order of this ACE for the associated entry as read from the file system
canonical_index	smallint	smallint	Preferred order in which ACE should appear for the associated entry

3.1.28 SRS.SCANS

Table 3-6 *Scans Table Definition*

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
id	bigint	bigint	Primary key
scan_policy_id	integer	integer	Reference to scan_policies table
triggered_start_time	datetime2(3)	timestamp without time zone	Initial time scan delegation starts
scan_start_time	datetime2(3)	timestamp without time zone	Start time when agent begins physical scan
scan_stop_time	datetime2(3)	timestamp without time zone	Stop time when agent completes physical scan
enum_start_time	datetime2(3)	timestamp without time zone	Agent metrics related to file system object enumeration
enum_stop_time	datetime2(3)	timestamp without time zone	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 without time zone	Metrics related to agent caching

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
caching_stop_time	datetime2(3)	timestamp without time zone	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 without time zone	Agent metrics related to filesystem metadata collection
metadata_stop_time	datetime2(3)	timestamp without time zone	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 without time zone	Agent metrics related to security principal collection
accounts_stop_time	datetime2(3)	timestamp without time zone	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 without time zone	Related to transfer of scan file from the Agent to the Engine
transfer_stop_time	datetime2(3)	timestamp without time zone	Related to transfer of scan file from the Agent to the Engine

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
db_start_time	datetime2(3)	timestamp without time zone	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 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending 6 = Previous 7 = Retained
next_retry_time	datetime2(0)	timestamp without time zone	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)	

* Database insert times do not include security equivalence, group membership, or eDirectory Directory Service trustee processing, all of which runs in the background.

3.1.29 SRS.SCAN_DATA

Table 3-7 *Scan Data Table Definition*

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
id	bigint	bigint	Primary key
scan_id	integer	integer	Reference to scans 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
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
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	Maps to either a GUID or a 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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
create_time	datetime2(0)	timestamp without time zone	
modify_time	datetime2(0)	timestamp without time zone	
access_time	datetime2(0)	timestamp without time zone	
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	

3.1.30 SRS.SCAN_DIRECTORY_DATA

Table 3-8 *Scan Directory Data Table Definition*

Column Name	SQL Server Data Type	PostgreSQL Data Type	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

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

3.1.31 SRS.SCAN_HISTORY

Table 3-9 *Scan History Table Definition*

Column Name	SQL Server Data Type	PostgreSQL Data Type	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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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 without time zone	Initial time scan delegation starts
scan_start_time	datetime2(3)	timestamp without time zone	Start time when agent begins physical scan
scan_stop_time	datetime2(3)	timestamp without time zone	Stop time when agent completes physical scan
enum_start_time	datetime2(3)	timestamp without time zone	Agent metrics related to file system object enumeration
enum_stop_time	datetime2(3)	timestamp without time zone	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 without time zone	Metrics related to agent caching
caching_stop_time	datetime2(3)	timestamp without time zone	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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
cache_size_max	integer	integer	Metrics related to agent caching
metadata_start_time	datetime2(3)	timestamp without time zone	Agent metrics related to filesystem metadata collection
metadata_stop_time	datetime2(3)	timestamp without time zone	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 without time zone	Agent metrics related to security principal collection
accounts_stop_time	datetime2(3)	timestamp without time zone	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 without time zone	Related to transfer of scan file from the Agent to the Engine
transfer_stop_time	datetime2(3)	timestamp without time zone	Related to transfer of scan file from the Agent to the Engine
db_start_time	datetime2(3)	timestamp without time zone	Database insert start time*
db_stop_time	datetime2(3)		Database insert stop time*
status_code	integer	integer	Internal status code
error_string	nvarchar(1024)	varchar(1024)	

3.1.32 SRS.SCAN_TARGETS

Table 3-10 *Scan Targets Table Definition*

Column Name	SQL Server Data Type	PostgreSQL Data Type	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

3.1.33 SRS.SECURITY_DESCRIPTORS

Table 3-11 *Security Descriptors Table Definition*

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
id	bigint	bigint	Primary key
scan_data_id	bigint	bigint	Reference to scan data table
control	integer	integer	Security descriptor control flags
dacl_present	bit	boolean	Indicates presence of DACL entries for this security descriptor
sacl_present	bit	boolean	Indicates presence of SACL entries for this security descriptor

3.1.34 SRS.TREND_VOLUME_FREESPACE

Table 3-12 *Trend Volume Freespace Table Definition*

Column Name	SQL Server Data Type	PostgreSQL Data Type	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 without time zone	

3.2 Views

- Section 3.2.11, “SRS.BASELINE_NTFS_ACES,” on page 71
- Section 3.2.12, “SRS.BASELINE_PERMISSIONS_SCANS,” on page 74

3.2.1 SRS.CURRENT_FS_SCANDATA

Table 3-13 *Current File System Scan Data Unified View*

Column Name	SQL Server Data Type	PostgreSQL Data Type	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 without time zone	Stored as UTC time
modify_time	datetime2(0)	timestamp without time zone	Stored as UTC time
access_time	datetime2(0)	timestamp without time zone	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
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)

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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
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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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	

3.2.2 SRS.CURRENT_FS_SCANS

Table 3-14 *Current File System Scans View*

Column Name	SQL Server Data Type	PostgreSQL Data Type	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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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
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

3.2.3 SRS.CURRENT_NTFS_ACES

Table 3-15 *Current NTFS ACEs View*

Column Name	SQL Server Data Type	PostgreSQL Data Type	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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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 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 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

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

3.2.4 SRS.CURRENT_PERMISSIONS_SCANS

Table 3-16 *Current Permissions Scans View*

Column Name	SQL Server Data Type	PostgreSQL Data Type	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 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

3.2.5 SRS.PREVIOUS_FS_SCANDATA

Table 3-17 *Previous File System Scan Data Unified View*

Column Name	SQL Server Data Type	PostgreSQL Data Type	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 without time zone	Stored as UTC time
modify_time	datetime2(0)	timestamp without time zone	Stored as UTC time
access_time	datetime2(0)	timestamp without time zone	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
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)

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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
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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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	

3.2.6 SRS.PREVIOUS_FS_SCANS

Table 3-18 *Previous File System Scans View*

Column Name	SQL Server Data Type	PostgreSQL Data Type	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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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
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

3.2.7 SRS.PREVIOUS_NTFS_ACES

Table 3-19 *Previous NTFS ACEs View*

Column Name	SQL Server Data Type	PostgreSQL Data Type	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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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 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 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

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

3.2.8 SRS.PREVIOUS_PERMISSIONS_SCANS

Table 3-20 *Previous Permissions Scans View*

Column Name	SQL Server Data Type	PostgreSQL Data Type	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 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

3.2.9 SRS.BASELINE_FS_SCANDATA

Table 3-21 *Baseline File System Scan Data Unified View*

Column Name	SQL Server Data Type	PostgreSQL Data Type	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 without time zone	Stored as UTC time
modify_time	datetime2(0)	timestamp without time zone	Stored as UTC time
access_time	datetime2(0)	timestamp without time zone	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
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)

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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
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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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	

3.2.10 SRS.BASELINE_FS_SCANS

Table 3-22 *Baseline File System Scans View*

Column Name	SQL Server Data Type	PosgreSQL Data Type	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

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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
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

3.2.11 SRS.BASELINE_NTFS_ACES

Table 3-23 *Baseline NTFS ACEs View*

Column Name	SQL Server Data Type	PosgreSQL Data Type	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

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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 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 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

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
ace_flags	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 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
ad_object_id	integer	integer	Reference to ad_objects table

3.2.12 SRS.BASELINE_PERMISSIONS_SCANS

Table 3-24 *Baseline Permissions Scans View*

Column Name	SQL Server Data Type	PosgreSQL Data Type	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 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

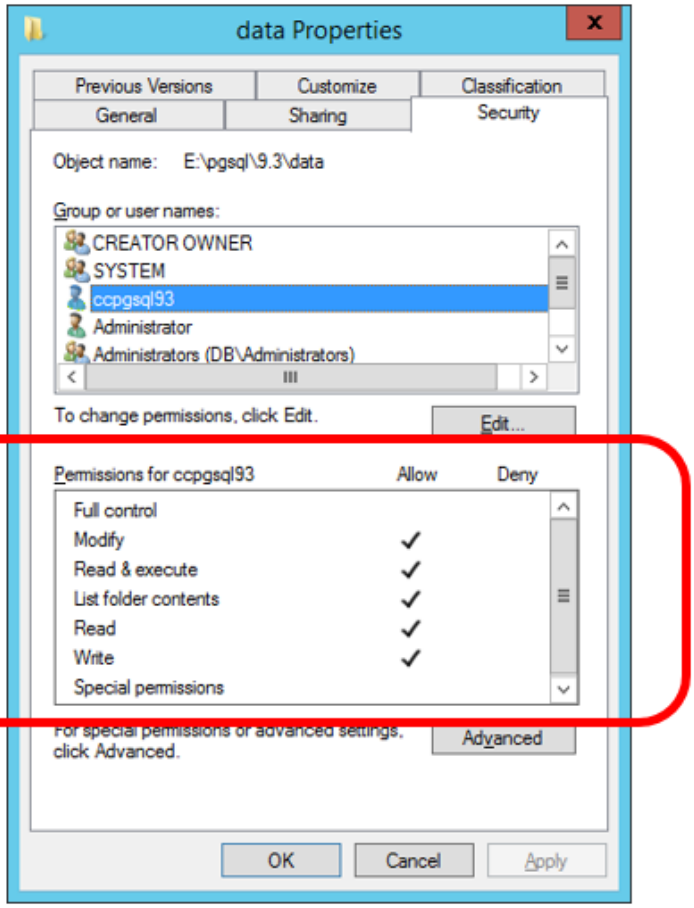
3.3 Functions

3.3.1 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)

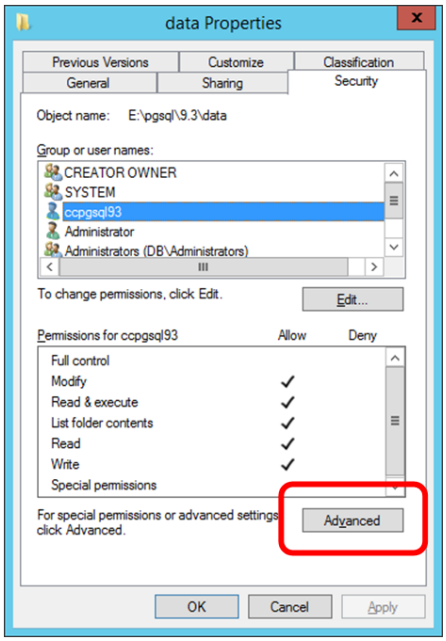
```
SELECT
    sd.fullpath,
    srs.access_mask_basic_string(ntfs.access_mask, 2) AS basic_permissions
FROM srs.ntfs_aces AS ntfs
JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id
WHERE sd.path_type = 2
LIMIT 100;
```

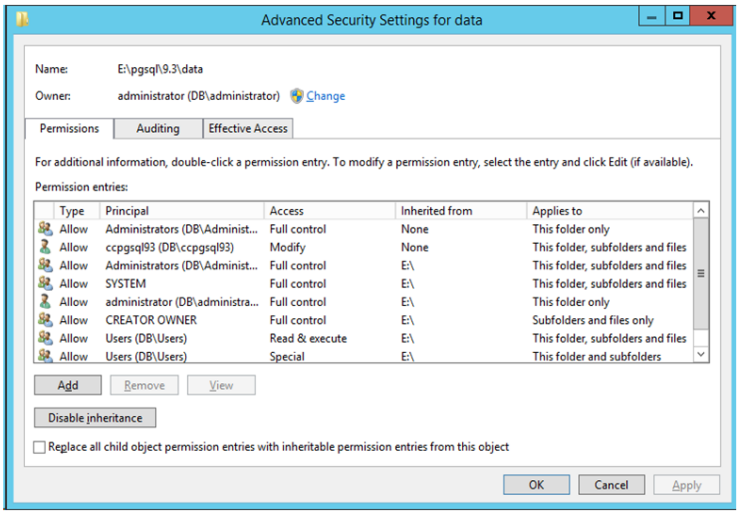
3.3.2 SRS.ACCESS_MASK_STRING

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 / Lf	Read data / List folder
0x00000002	Wd / Cf	Write data / Create file
0x00000004	Ad / Cs	Append data / Create subdirectory
0x00000008	Rx	Read extended attributes
0x00000010	Wx	Write extended attributes
0x00000020	Xf / Tf	File execute / Traverse
0x00000040	Ds	Delete child (subdirectory)
0x00000080	Ra	Read attributes
0x00000100	Wa	Write attributes
0x00010000	De	Delete
0x00020000	Rp	Read permissions
0x00040000	Cp	Change permissions
0x00080000	To	Change owner (take ownership)
0x00100000	Sy	Synchronize
0x01000000	Ss	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;
```

3.3.3 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 a 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

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

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

3.3.4 SRS.ACE_FLAGS_STRING

Parameters	SQL Server	PostgreSQL
@flags	integer	integer

Parameters	SQL Server	PostgreSQL
Return Value	nvarchar(128)	varchar(128)

Description: Converts the access mask flag 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 (ID) Inherited
0x040 (SA) Successful access
0x080 (FA) Failed access

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
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
FROM srs.ntfs_aces AS ntfs
JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id
LIMIT 100;
```

3.3.5 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;
```

3.3.6 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)

```
SELECT TOP(100) fullpath, srs.attribute_string(attributes) FROM srs.scan_data;
```

Example (PostgreSQL)

```
SELECT fullpath, srs.attribute_string(attributes) FROM srs.scan_data LIMIT 100;
```

3.3.7 SRS.BYTE_STRING

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

Description: Converts a number 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

```
SELECT srs.byte_string(1287168)
```

3.3.8 SRS.BYTE_UNIT_STRING

Parameters	SQL Server	PostgreSQL
@size	bigint	bigint
@unit	nvarchar(10)	text
@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

```
SELECT srs.byte_unit_string(1287201, 'KB', 3)
```

3.3.9 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

```
SELECT
    srs.bytes_to_hex_string(ad.sid)
FROM srs.ad_objects AS ad
```

3.3.10 SRS.HEX_STRING_TO_BYTES

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

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

- 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 '0's are required for each hex value having a single digit.
- Do not include separators such as '-' between hex values.

Example

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

3.3.11 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 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

```
SELECT srs.guid_bytes('{12345678-1234-5678-9abc-123456789abc}')
```

3.3.12 SRS.GUID_TEXT

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

Description: Converts a binary guid value to its equivalent string representation. Note that returned guid strings are in the format {*XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX*}. All hex values are returned with uppercase A-F.

Example

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

3.3.13 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.
- Useful for finding a fullpath in the srs.scan_data table using the fullpath_hash index.

Example

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

3.3.14 SRS.SID_BYTES

Parameters	SQL Server	PostgreSQL
@sid	nvarchar(1024)	varchar(1024)
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

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

3.3.15 SRS.SID_TEXT

Parameters	SQL Server	PostgreSQL
@sid_bytes	varbinary(68)	bytea
Return Value	nvarchar(1024)	varchar(1024)

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

Example

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