



# Micro Focus File Reporter 3.6 Database Schema and Custom Queries Guide

November 2,7 2018

## 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.

Copyright © 2018 Condrey Corporation. All Rights Reserved.

No part of this publication may be reproduced, photocopied, or transmitted in any fashion with out the express written consent of the publisher.

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

<b>About This Guide</b>	<b>5</b>
<b>1 Overview</b>	<b>7</b>
1.1 Updates and Breaking Changes	7
1.1.1 Additional scan_id Column	7
1.1.2 Added srs.byte_unit_string Function	7
1.1.3 Deprecated Column is_valid	7
1.1.4 Active Views	8
1.2 Supported Schema Objects	8
1.3 Schema Namespace	8
1.4 Supported Database Tables	9
1.5 Supported Database Views	9
1.6 Supported Functions	10
<b>2 Custom Query</b>	<b>13</b>
2.1 Understanding Table Relationships	13
2.1.1 File System Data	13
2.1.2 OES File System Permissions Overview	14
2.1.3 NTFS Permissions Overview	15
2.2 Scoping and Filtering	16
2.2.1 Scope by Identity System	16
2.2.2 Scope by Server	17
2.2.3 Scope by Scan Target	17
2.2.4 Scope by Directory	18
2.2.5 Scope by Directory with Path Depth Limit	19
2.2.6 Scope by Security Principal	20
2.2.7 Basic Filtering	21
<b>3 Custom Schema Reference</b>	<b>25</b>
3.1 Tables	25
3.1.1 ANALYSIS.FILE_SCAN_ENTRIES	25
3.1.2 SRS.AD_MEMBERSHIPS	26
3.1.3 SRS.AD_OBJECTS	26
3.1.4 SRS.EDIR_DS_TRUSTEES	27
3.1.5 SRS.EDIR_OBJECTS	27
3.1.6 SRS.EDIR_SECURITY_EQUALS	28
3.1.7 SRS.IDENTITY_SYSTEMS	28
3.1.8 SRS.NCP_TRUSTEES	29
3.1.9 SRS.NTFS_ACES	29
3.1.10 SRS.SCANS	31
3.1.11 SRS.SCAN_DATA	33
3.1.12 SRS.SCAN_DIRECTORY_DATA	36
3.1.13 SRS.SCAN_HISTORY	37
3.1.14 SRS.SCAN_TARGETS	39
3.1.15 SRS.SECURITY_DESCRIPTOR	40
3.1.16 SRS.TREND_VOLUME_FREESPACE	40
3.2 Views	41
3.2.1 SRS.CURRENT_FS_SCANDATA	42

3.2.2	SRS.CURRENT_FS_SCANDATA_AD	44
3.2.3	SRS.CURRENT_FS_SCANDATA_EDIR	46
3.2.4	SRS.CURRENT_FS_SCANS	48
3.2.5	SRS.CURRENT_NCP_TRUSTEES	50
3.2.6	SRS.CURRENT_NTFS_ACES	51
3.2.7	SRS.CURRENT_PERMISSIONS_SCANS	54
3.2.8	SRS.PREVIOUS_FS_SCANDATA	56
3.2.9	SRS.PREVIOUS_FS_SCANDATA_AD	58
3.2.10	SRS.PREVIOUS_FS_SCANDATA_EDIR	60
3.2.11	SRS.PREVIOUS_FS_SCANS	62
3.2.12	SRS.PREVIOUS_NCP_TRUSTEES	64
3.2.13	SRS.PREVIOUS_NTFS_ACES	65
3.2.14	SRS.PREVIOUS_PERMISSIONS_SCANS	68
3.2.15	SRS.BASELINE_FS_SCANDATA	70
3.2.16	SRS.BASELINE_FS_SCANDATA_AD	72
3.2.17	SRS.BASELINE_FS_SCANDATA_EDIR	74
3.2.18	SRS.BASELINE_FS_SCANS	76
3.2.19	SRS.BASELINE_NCP_TRUSTEES	78
3.2.20	SRS.BASELINE_NTFS_ACES	79
3.2.21	SRS.BASELINE_PERMISSIONS_SCANS	82
3.3	Functions	83
3.3.1	SRS.ACCESS_MASK_BASIC_STRING	84
3.3.2	SRS.ACCESS_MASK_STRING	86
3.3.3	SRS.AD_ACCOUNT_NAME	87
3.3.4	SRS.ACE_FLAGS_STRING	88
3.3.5	SRS.ACE_TYPE_STRING	89
3.3.6	SRS.ATTRIBUTE_STRING	90
3.3.7	SRS.BYTE_STRING	91
3.3.8	SRS.BYTE_UNIT_STRING	91
3.3.9	SRS.GUID_BYTES	92
3.3.10	SRS.GUID_TEXT	92
3.3.11	SRS.NCP_RIGHTS_STRING	93
3.3.12	SRS.PATH_HASH	94
3.3.13	SRS.SID_BYTES	94
3.3.14	SRS.SID_TEXT	95

**A Documentation Updates 97**

A.1	July 2, 2018	97
A.2	July 19, 2016	97
A.3	April 27, 2015	97

# 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.

- ◆ [Chapter 1, “Overview,” on page 7](#)
- ◆ [Chapter 2, “Custom Query,” on page 13](#)
- ◆ [Chapter 3, “Custom Schema Reference,” on page 25](#)
- ◆ [Appendix A, “Documentation Updates,” on page 97](#)

## Audience

This manual is intended for database administrators who want to generate Custom Query reports using File Reporter 3.6.

## 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](http://www.novell.com/documentation/feedback.html) and enter your comments there.

## Documentation Updates

For the most recent version of the *Micro Focus File Reporter 3.6 Database Schema and Custom Queries Guide*, visit the [Micro Focus File Reporter Documentation Web site \(http://www.novell.com/documentation/filereporter3\)](http://www.novell.com/documentation/filereporter3).

## Additional Documentation

For additional Micro Focus File Reporter documentation, see the following guides at the [Micro Focus File Reporter Documentation Web site \(http://www.novell.com/documentation/filereporter3\)](http://www.novell.com/documentation/filereporter3):

- ◆ *Micro Focus File Reporter 3.6 Installation Guide*
- ◆ *Micro Focus File Reporter 3.6 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

- ◆ Section 1.1, “Updates and Breaking Changes,” on page 7
- ◆ Section 1.2, “Supported Schema Objects,” on page 8
- ◆ Section 1.3, “Schema Namespace,” on page 8
- ◆ Section 1.4, “Supported Database Tables,” on page 9
- ◆ Section 1.5, “Supported Database Views,” on page 9
- ◆ Section 1.6, “Supported Functions,” on page 10

## 1.1 Updates and Breaking Changes

- ◆ Section 1.1.1, “Additional scan\_id Column,” on page 7
- ◆ Section 1.1.2, “Added srs.byte\_unit\_string Function,” on page 7
- ◆ Section 1.1.3, “Deprecated Column is\_valid,” on page 7
- ◆ Section 1.1.4, “Active Views,” on page 8

### 1.1.1 Additional scan\_id Column

The `scan_id` column has now been extended to the following tables and indexed to provide simpler joins and quicker lookup:

- ◆ `srs.scan_directory_data`
- ◆ `srs.ntfs_aces`
- ◆ `srs.security_descriptors`
- ◆ `srs.ncp_trustees`

### 1.1.2 Added srs.byte\_unit\_string Function

The function `srs.byte_unit_string` was added in this release. This function is similar to `srs.byte_string` but allows the caller to specify which unit the formatted result will be displayed in, such as KB, MB, or GB.

### 1.1.3 Deprecated Column is\_valid

The column `is_valid` in the table `srs.scans` and the views `srs.current_fs_scans` and `srs.current_permissions_scans` is now deprecated. This column is redundant in that it reflects whether the `progress_status` column contains the value 4, indicating whether a scan is Current. Since the `progress_status` column maintains the scan state for the Current, Previous, and Retained scan statuses as well as other scan states, it should be used instead.

This field still remains in the schema for this release, but might be removed in a later version.

## 1.1.4 Active Views

The previously deprecated `srs.active_*` views have been removed in this release. Please update any queries referencing these to the equivalent replacement views listed below.

*Table 1-1 Active Views*

Removed View	Replacement View
<code>srs.active_fs_scans</code>	<code>srs.current_fs_scans</code>
<code>srs.active_fs_scandata</code>	<code>srs.current_fs_scandata</code>
<code>srs.active_fs_scandata_ad</code>	<code>srs.current_fs_scandata_ad</code>
<code>srs.active_fs_scandata_edir</code>	<code>srs.current_fs_scandata_edir</code>
<code>srs.active_permissions_scans</code>	<code>srs.current_permissions_scans</code>
<code>srs.active_ncp_trustees</code>	<code>srs.current_ncp_trustees</code>
<code>srs.active_ntfs_aces</code>	<code>srs.current_ntfs_aces</code>

## 1.2 Supported Schema Objects

The supported database schema objects are divided into the following categories:

- ◆ Identity Systems
- ◆ Scan Info
- ◆ File System Metadata
- ◆ File System Permissions Data

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

---

**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.

---

## 1.3 Schema Namespace

All supported database objects and functions reside in the `srs` schema namespace. 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 the supported database objects and functions with the `srs` prefix.



## 1.4 Supported Database Tables

Table 1-2 Supported Database Tables

Category	Table Name	Notes
Identity Systems	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.edir_objects	List of all scanned eDirectory security principals
	srs.edir_security_equals	eDirectory security equivalence relations
	srs.edir_ds_trustees	eDirectory objects with rights that flow into the file system
Scan Info	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
File System Metadata	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 freespace records
File System Permissions Data	srs.ncp_trustees	Scanned OES file system rights
	srs.ntfs_aces	Scanned NTFS ACEs
	srs.security_descriptors	Scanned NTFS security descriptors

## 1.5 Supported Database Views

Table 1-3 Supported Database Views

Category	View Name	Notes
Scan Info	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
File System Metadata	srs.current_fs_scandata_ad	All file system scan data from Current scans in Active Directory environments

Category	View Name	Notes
	srs.current_fs_scandata_edir	All file system scan data from Current scans in eDirectory 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_edir	All file system scan data from Previous scans in eDirectory 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_edir	All file system scan data from Baseline scans in eDirectory environments
	srs.baseline_fs_scandata	Combined list of all Baseline file system scan data
File System Permissions Data	srs.current_ncp_trustees	All Current permissions scan data in eDirectory environments
	srs.current_ntfs_aces	All Current permissions scan data in Active Directory environments
	srs.previous_ncp_trustees	All Previous permissions scan data in eDirectory environments
	srs.previous_ntfs_aces	All Previous permissions scan data in Active Directory environments
	srs.baseline_ncp_trustees	All Baseline permissions scan data in eDirectory environments
	srs.baseline_ntfs_aces	All Baseline permissions scan data in Active Directory environments

**IMPORTANT:** The generalized view `srs.current_fs_scandata` is a combination of the `srs.current_fs_scandata_edir` and `srs.current_fs_scandata_ad` views. This view is provided as a convenience, but its use might be significantly slower in some cases as opposed to getting the results on the more specific `*_ad` and `*_edir` views.

The same applies to the views `srs.previous_fs_scandata` and `srs.baseline_fs_scandata`.

## 1.6 Supported Functions

*Table 1-4 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

Category	View Name	Description
	srs.attribute_string	Converts attributes to string representation
	srs.guid_bytes	Converts Guid from string to binary
	srs.guid_text	Converts Guid from binary to string
	srs.path_hash	Calculates SHA-1 hash of 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
	srs.ncp_rights_string	Translates NCP trustee rights to string representation



# 2 Custom Query

- ◆ [Section 2.1, “Understanding Table Relationships,” on page 13](#)
- ◆ [Section 2.2, “Scoping and Filtering,” on page 16](#)

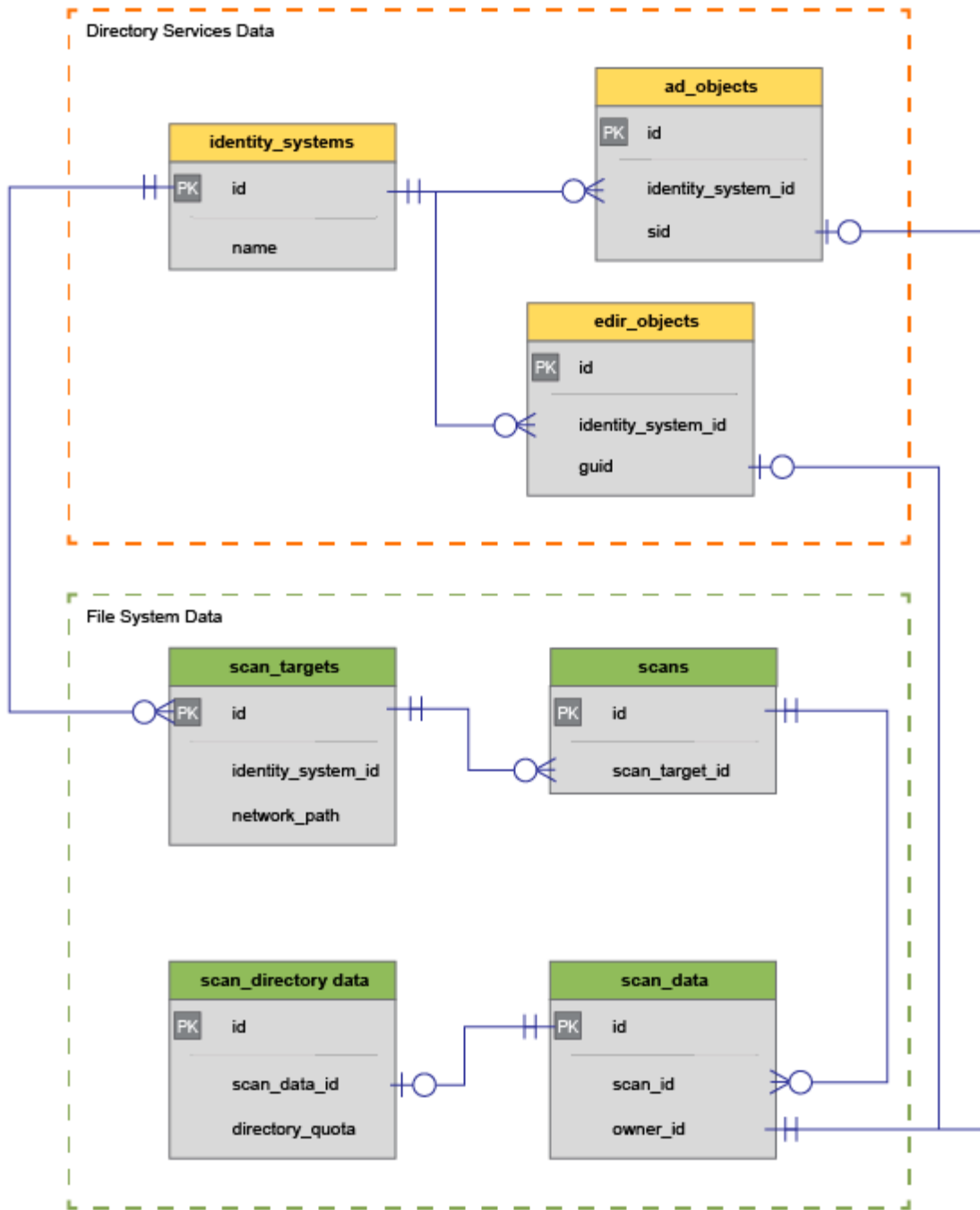
## 2.1 Understanding Table Relationships

- ◆ [Section 2.1.1, “File System Data,” on page 13](#)
- ◆ [Section 2.1.2, “OES File System Permissions Overview,” on page 14](#)
- ◆ [Section 2.1.3, “NTFS Permissions Overview,” on page 15](#)

### 2.1.1 File System Data

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

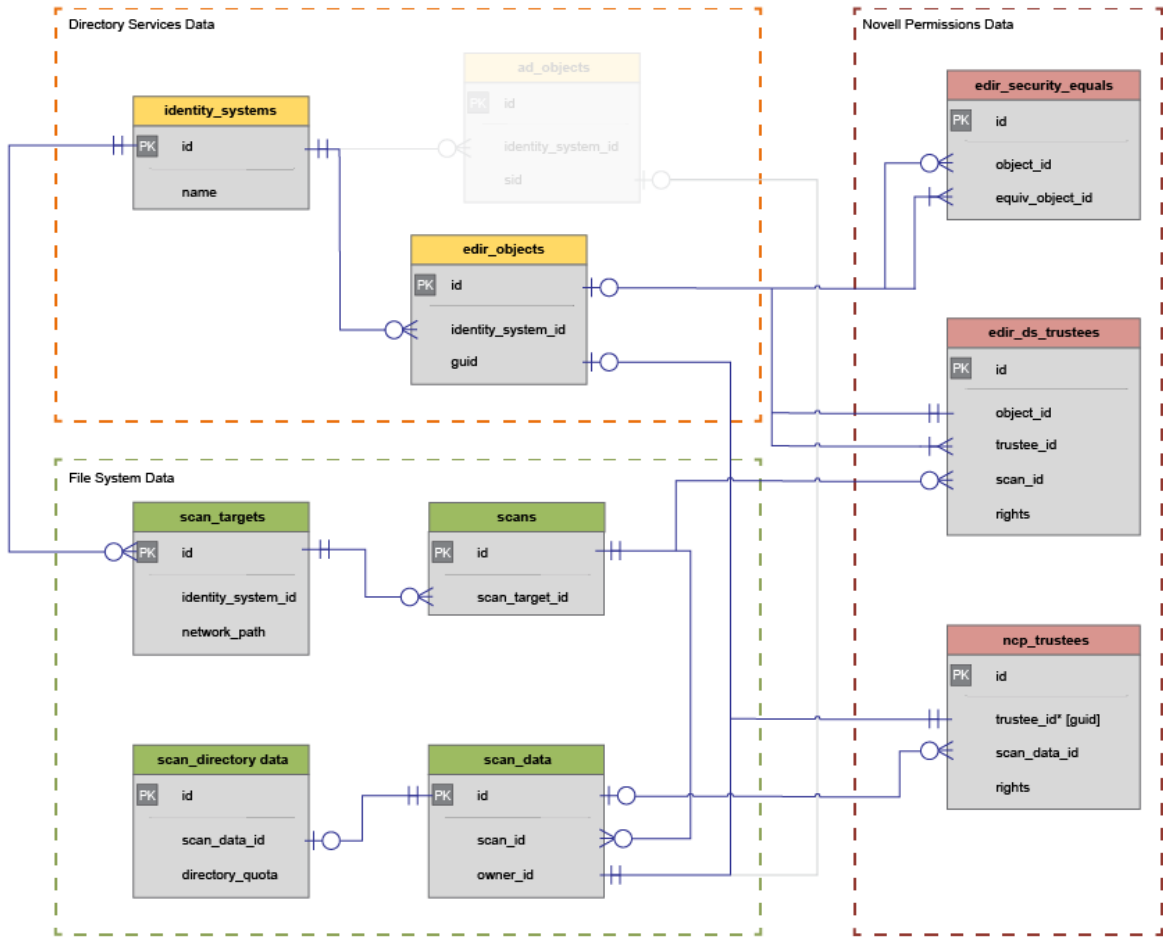
For general file system metadata collection, only file system data is collected, along with minimal identity system data pertaining to file and folder owners.



## 2.1.2 OES File System Permissions Overview

OES File System Permissions data is limited to folder structure as well as assigned and inherited NCP trustee rights.

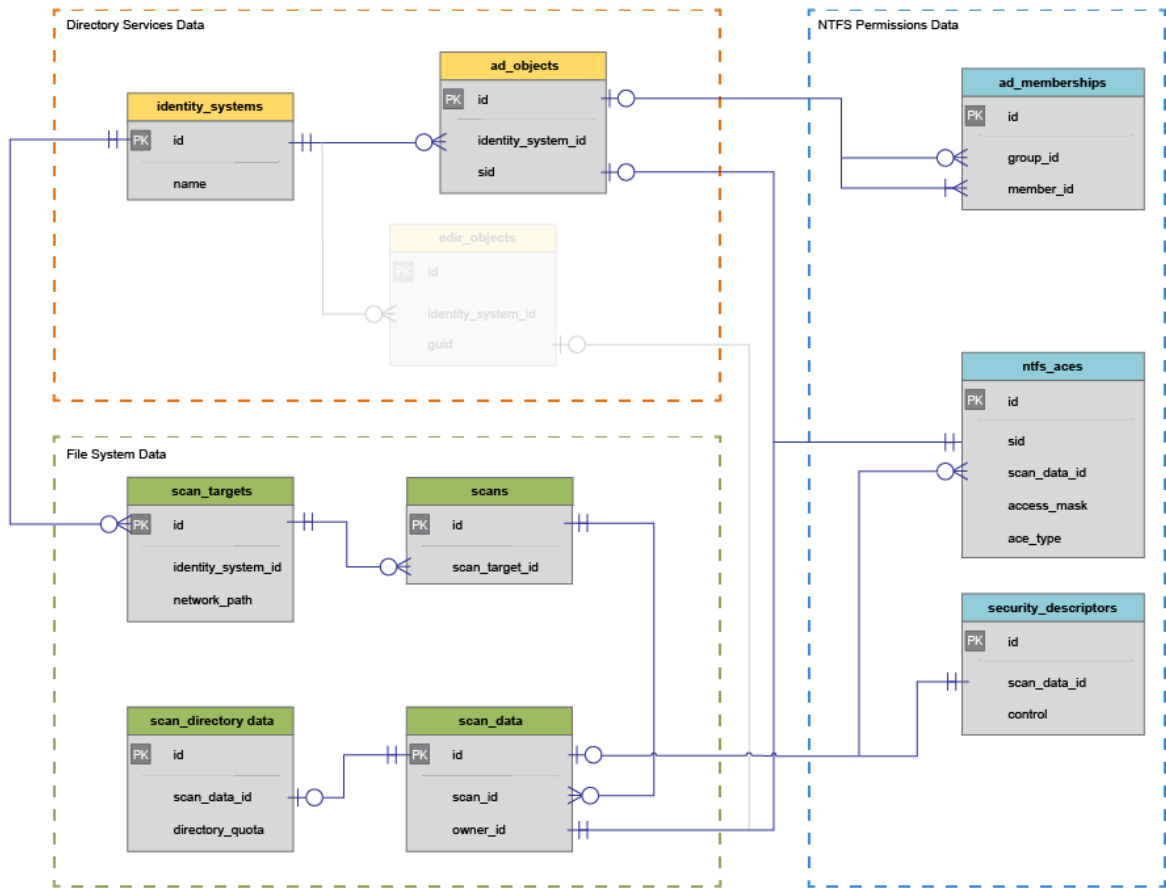
It should be noted that permissions scans do not include metadata specific information such as directory quota, nor do they include any file-entry data that is not a folder. Only permissions for folder entries are currently collected.



### 2.1.3 NTFS Permissions Overview

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 data in a given 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 2.2.1, “Scope by Identity System,” on page 16](#)
- [Section 2.2.2, “Scope by Server,” on page 17](#)
- [Section 2.2.3, “Scope by Scan Target,” on page 17](#)
- [Section 2.2.4, “Scope by Directory,” on page 18](#)
- [Section 2.2.5, “Scope by Directory with Path Depth Limit,” on page 19](#)
- [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
<code>idx</code>	Entry index.	Unique per scan.
<code>parent_idx</code>	Index of parent directory, share or DFS name space entry.	For all sibling file system entries, they will have the same parent index.
<code>path_depth</code>	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 <code>path_type</code> .  Entries occurring above the root path (such as DFS name spaces) will have a negative value.
<code>ns_left</code> , <code>ns_right</code>	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.

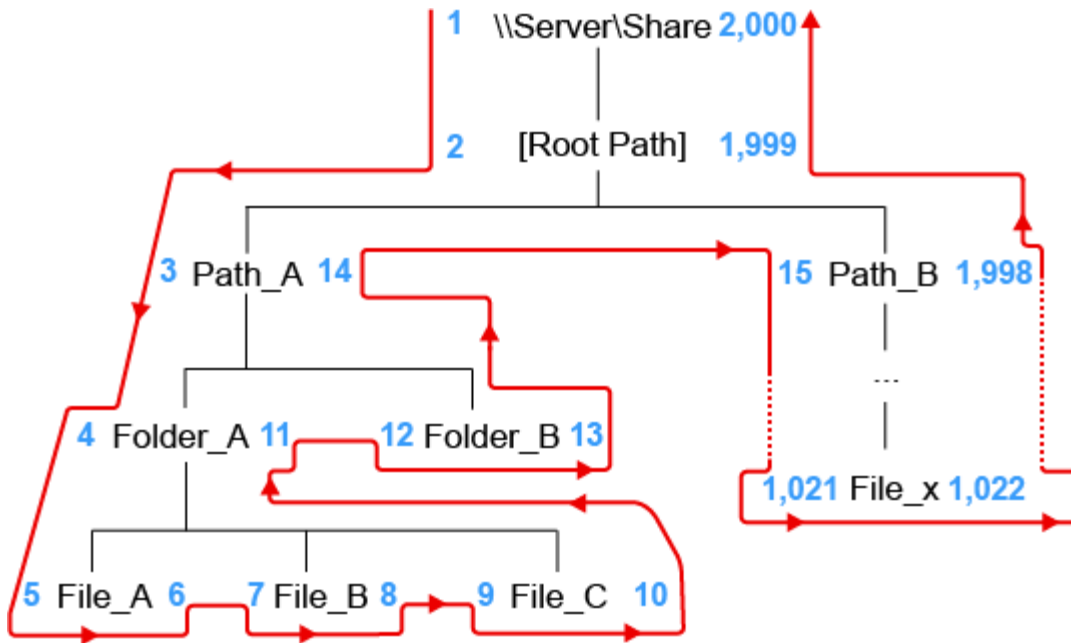
**Example:** Select all NTFS file system entries subordinate to, and including the specified target path.

```
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](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 22](#)
- ♦ [“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

- ◆ Section 3.1, “Tables,” on page 25
- ◆ Section 3.2, “Views,” on page 41
- ◆ Section 3.3, “Functions,” on page 83

### 3.1 Tables

- ◆ Section 3.1.1, “ANALYSIS.FILE\_SCAN\_ENTRIES,” on page 25
- ◆ Section 3.1.2, “SRS.AD\_MEMBERSHIPS,” on page 26
- ◆ Section 3.1.3, “SRS.AD\_OBJECTS,” on page 26
- ◆ Section 3.1.4, “SRS.EDIR\_DS\_TRUSTEES,” on page 27
- ◆ Section 3.1.5, “SRS.EDIR\_OBJECTS,” on page 27
- ◆ Section 3.1.6, “SRS.EDIR\_SECURITY\_EQUALS,” on page 28
- ◆ Section 3.1.7, “SRS.IDENTITY\_SYSTEMS,” on page 28
- ◆ Section 3.1.8, “SRS.NCP\_TRUSTEES,” on page 29
- ◆ Section 3.1.9, “SRS.NTFS\_ACES,” on page 29
- ◆ Section 3.1.10, “SRS.SCANS,” on page 31
- ◆ Section 3.1.11, “SRS.SCAN\_DATA,” on page 33
- ◆ Section 3.1.12, “SRS.SCAN\_DIRECTORY\_DATA,” on page 36
- ◆ Section 3.1.13, “SRS.SCAN\_HISTORY,” on page 37
- ◆ Section 3.1.14, “SRS.SCAN\_TARGETS,” on page 39
- ◆ Section 3.1.15, “SRS.SECURITY\_DESCRIPTOR,” on page 40
- ◆ Section 3.1.16, “SRS.TREND\_VOLUME\_FREESPACE,” on page 40

#### 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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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 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	

### 3.1.3 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
guid	binary(16)	bytea	Globally Unique Identifier

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
sid	varbinary(68)	bytea	Security Identifier
object_type	integer	integer	0 = Unknown / Other 1 = User 2 = Group 3 = Computer 4 = Container 5 = Built-in domain 6 = Organization 7 = Organizational Unit 8 = Locality 9 = Domain DNS 10 = Volume 13 = Inet Org Person 16 = Country 17 = Tree Root 18 = Dynamic Group 20 = Alias 21 = eDirectory Pseudo Security Principal
identity_system_id	integer	integer	Reference to primary key of identity_systems table

### 3.1.4 SRS.EDIR\_DS\_TRUSTEES

*Table 3-4 eDirectory DS Trustees Table Definition*

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
id	bigint	bigint	Primary key
scan_id	integer	integer	Reference to scans table
object_id	integer	integer	Reference to edir_objects table entry
trustee_id	integer	integer	Reference to edir_objects table entry
rights	integer	integer	Assigned eDirectory DS rights

### 3.1.5 SRS.EDIR\_OBJECTS

*Table 3-5 eDirectory Objects Table Definition*

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
id	integer	integer	Primary key
name	nvarchar(256)	varchar(256)	Common name (CN)
fdn	nvarchar(512)	varchar(512)	Full distinguished name

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
guid	varbinary(68)	bytea	Globally unique Identifier
object_type	integer	integer	0 = Unknown / Other 1 = User 2 = Group 3 = Computer 4 = Container 5 = Built-in domain 6 = Organization 7 = Organizational Unit 8 = Locality 9 = Domain DNS 10 = Volume 13 = Inet Org Person 16 = Country 17 = Tree Root 18 = Dynamic Group 20 = Alias 21 = eDirectory Pseudo Security Principal
identity_system_id	integer	integer	Reference to identity_systems table

### 3.1.6 SRS.EDIR\_SECURITY\_EQUALS

*Table 3-6 eDirectory Security Equals Table Definition*

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
id	bigint	bigint	Primary key
object_id	integer	integer	Reference to edir_objects table
equiv_object_id	integer	integer	Reference to edir_objects table

### 3.1.7 SRS.IDENTITY\_SYSTEMS

*Table 3-7 Identity Systems Table Definition*

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
id	integer	integer	Primary key
type	integer	integer	0 = Unknown 1 = Active Directory 2 = eDirectory 3 = Windows Local

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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	

### 3.1.8 SRS.NCP\_TRUSTEES

*Table 3-8 NCP Trustees Table Definition*

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
id	bigint	bigint	Primary key
scan_data_id	bigint	bigint	Reference to scan_data table
trustee_guid	varbinary(68)	bytea	Reference to edir_objects (using GUID)
rights	integer	integer	0x01 = Read 0x02 = Write 0x04 = [undefined] 0x08 = Create 0x10 = Delete 0x20 = Access Control 0x40 = File Scan 0x80 = Modify 0x100 = Supervisor

### 3.1.9 SRS.NTFS\_ACES

*Table 3-9 NTFS ACEs Table Definition*

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
id	bigint	bigint	Primary key

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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 Permissions 0x80000 = Change Owner 0x100000 = Synchronize 0x1000000 = Access System Security 0x10000000 = Generic All 0x20000000 = Generic Execute 0x40000000 = Generic Write 0x80000000 = Generic Read
sid	varbinary(68)	bytea	Trustee SID
index_on_disk	smallint	smallint	Discovered order of this ACE for the associated entry as read from the file system

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
canonical_index	smallint	smallint	Preferred order in which ACE should appear for the associated entry

### 3.1.10 SRS.SCANS

*Table 3-10 Scans Table Definition*

Column Name	SQL Server Data Type	PosgreSQL 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
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

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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
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	



Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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.

### 3.1.11 SRS.SCAN\_DATA

*Table 3-11 Scan Data Table Definition*

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
id	bigint	bigint	Primary key
scan_id	integer	integer	Reference to scans table

Column Name	SQL Server Data Type	PosgreSQL 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
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

Column Name	SQL Server Data Type	PosgreSQL 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 0x8000 = NSS Execute only 0x10000 = NSS Shareable 0x20000 = NSS Don't Sub-allocate 0x40000 = NSS Transactional 0x80000 = NSS Indexed 0x100000 = NSS Read Audit 0x200000 = NSS Write Audit 0x400000 = NSS Immediate Purge 0x800000 = NSS Rename Inhibit 0x1000000 = NSS Delete Inhibit 0x2000000 = NSS Copy Inhibit 0x4000000 = NSS Don't Migrate 0x8000000 = NSS Immediate Compress 0x10000000 = NSS Don't Compress 0x20000000 = NSS Can't Compress 0x40000000 = NSS File Audit 0x80000000 = NSS Is a Link
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

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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	

### 3.1.12 SRS.SCAN\_DIRECTORY\_DATA

*Table 3-12 Scan Directory Data Table Definition*

Column Name	SQL Server Data Type	PosgreSQL 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
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)

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
child_link_size	bigint	bigint	Size of all immediately subordinate links

### 3.1.13 SRS.SCAN\_HISTORY

*Table 3-13 Scan History Table Definition*

Column Name	SQL Server Data Type	PosgreSQL 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
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

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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
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

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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)	

### 3.1.14 SRS.SCAN\_TARGETS

*Table 3-14 Scan Targets Table Definition*

Column Name	SQL Server Data Type	PosgreSQL 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 2 = Linux 3 = Open Enterprise Server
filesystem	smallint	smallint	0 = Unknown 1 = NTFS 3 = NSS 4 = TFS 5 = Ext3 6 = ReiserFS 10 = Ext2 11 = Ext4 12 = Xfs 13 = Ramdisk 14 = OES_VFS 16 = Linux_generic 17 = NFS
cost_per_unit	money	money	Not currently used

### 3.1.15 SRS.SECURITY\_DESCRIPTOR

*Table 3-15 Security Descriptors Table Definition*

Column Name	SQL Server Data Type	PosgreSQL 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

### 3.1.16 SRS.TREND\_VOLUME\_FREESPACE

*Table 3-16 Trend Volume Freespace Table Definition*

Column Name	SQL Server Data Type	PosgreSQL 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 3 = NSS 4 = TFS 5 = Ext3 6 = ReiserFS 10 = Ext2 11 = Ext4 12 = Xfs 13 = Ramdisk 14 = OES_VFS 16 = Linux_generic 17 = NFS
volume_guid	uniqueidentifier	uuid	
volume_label	nvarchar(256)	text	
volume_bytes_total	bigint	bigint	
volume_bytes_free	bigint	bigint	



Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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 3.2.1, “SRS.CURRENT\\_FS\\_SCANDATA,” on page 42](#)
- ◆ [Section 3.2.2, “SRS.CURRENT\\_FS\\_SCANDATA\\_AD,” on page 44](#)
- ◆ [Section 3.2.3, “SRS.CURRENT\\_FS\\_SCANDATA\\_EDIR,” on page 46](#)
- ◆ [Section 3.2.4, “SRS.CURRENT\\_FS\\_SCANS,” on page 48](#)
- ◆ [Section 3.2.5, “SRS.CURRENT\\_NCP\\_TRUSTEES,” on page 50](#)
- ◆ [Section 3.2.6, “SRS.CURRENT\\_NTFS\\_ACES,” on page 51](#)
- ◆ [Section 3.2.7, “SRS.CURRENT\\_PERMISSIONS\\_SCANS,” on page 54](#)
- ◆ [Section 3.2.8, “SRS.PREVIOUS\\_FS\\_SCANDATA,” on page 56](#)
- ◆ [Section 3.2.9, “SRS.PREVIOUS\\_FS\\_SCANDATA\\_AD,” on page 58](#)
- ◆ [Section 3.2.10, “SRS.PREVIOUS\\_FS\\_SCANDATA\\_EDIR,” on page 60](#)
- ◆ [Section 3.2.11, “SRS.PREVIOUS\\_FS\\_SCANS,” on page 62](#)
- ◆ [Section 3.2.12, “SRS.PREVIOUS\\_NCP\\_TRUSTEES,” on page 64](#)
- ◆ [Section 3.2.13, “SRS.PREVIOUS\\_NTFS\\_ACES,” on page 65](#)
- ◆ [Section 3.2.14, “SRS.PREVIOUS\\_PERMISSIONS\\_SCANS,” on page 68](#)
- ◆ [Section 3.2.15, “SRS.BASELINE\\_FS\\_SCANDATA,” on page 70](#)
- ◆ [Section 3.2.16, “SRS.BASELINE\\_FS\\_SCANDATA\\_AD,” on page 72](#)
- ◆ [Section 3.2.17, “SRS.BASELINE\\_FS\\_SCANDATA\\_EDIR,” on page 74](#)
- ◆ [Section 3.2.18, “SRS.BASELINE\\_FS\\_SCANS,” on page 76](#)
- ◆ [Section 3.2.19, “SRS.BASELINE\\_NCP\\_TRUSTEES,” on page 78](#)
- ◆ [Section 3.2.20, “SRS.BASELINE\\_NTFS\\_ACES,” on page 79](#)
- ◆ [Section 3.2.21, “SRS.BASELINE\\_PERMISSIONS\\_SCANS,” on page 82](#)

### 3.2.1 SRS.CURRENT\_FS\_SCANDATA

Table 3-17 Current File System Scan Data Unified View

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
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 for Active Directory Common Name (CN) for eDirectory
owner_fdn	nvarchar(512)	varchar(512)	Full distinguished object name
owner_display_name	nvarchar(max)	text	For Active Directory – <i>DOMAIN\SamAccountName</i> For eDirectory – FDN
owner_id	varbinary(68)	bytea	SID for Active Directory GUID for eDirectory

Column Name	SQL Server Data Type	PosgreSQL 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	PosgreSQL 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\_SCANDATA\_AD

*Table 3-18 Current File System Scan Data for Active Directory View*

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
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			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 zone
modify_time	datetime2(0)	timestamp without time zone	Stored as UTC time zone
access_time	datetime2(0)	timestamp without time zone	Stored as UTC time zone

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
size	bigint	bigint	For files, actual size; for directories, accumulative size of all subordinate files
size_on_disk	bigint	bigint	Assumes typical allocation unit size of 4K
size_compressed	bigint	bigint	Only accurate for NTFS file systems
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	<i>DOMAIN\SAMAccount</i>
owner_id	varbinary(68)	bytea	SID
ad_object_id	integer	integer	Reference to ad_objects table
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

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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	

### 3.2.3 SRS.CURRENT\_FS\_SCANDATA\_EDIR

*Table 3-19 Current File System Scan Data for eDirectory View*

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name <ul style="list-style-type: none"> <li>♦ Active Directory – Forest FDN</li> <li>♦ eDirectory – Tree name</li> </ul>
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

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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_name	nvarchar(256)	varchar(256)	Common Name (CN)
owner_fdn	nvarchar(512)	varchar(512)	Full distinguished object name
owner_display_name	nvarchar(max)	text	Full distinguished object name
owner_id	varbinary(68)	bytea	GUID
edir_object_id	integer	integer	Reference to edir_objects table
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

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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	

### 3.2.4 SRS.CURRENT\_FS\_SCANS

*Table 3-20 Current 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 <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
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



Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
platform	integer	integer	0 = Unknown 1 = Windows 2 = Linux 3 = Open Enterprise Server
filesystem	integer	integer	0 = Unknown 1 = NTFS 3 = NSS 4 = TFS 5 = Ext3 6 = ReiserFS 10 = Ext2 11 = Ext4 12 = Xfs 13 = Ramdisk 14 = OES_VFS 16 = Linux_generic 17 = NFS
scan_type	integer	integer	Should always be 1
progress_status	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending 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

### 3.2.5 SRS.CURRENT\_NCP\_TRUSTEES

Table 3-21 Current NCP Trustees View

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
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_name	nvarchar(256)	varchar(256)	Common Name (CN)
trustee_fdn	nvarchar(512)	varchar(512)	Trustee's full distinguished name
trustee_display_name	nvarchar(512)	varchar(512)	Trustee's full distinguished name
trustee_type	integer	integer	0 = Unknown / Other 1 = User 2 = Group 3 = Computer 4 = Container 5 = Built-in domain 6 = Organization 7 = Organizational Unit 8 = Locality 9 = Domain DNS 10 = Volume 13 = Inet Org Person 16 = Country 17 = Tree Root 18 = Dynamic Group 20 = Alias 21 = eDirectory Pseudo Security Principal
trustee_guid	varbinary(68)	bytea	GUID
rights	integer	integer	0x01 = Read 0x02 = Write 0x04 = [undefined] 0x08 = Create 0x10 = Delete 0x20 = Access Control 0x40 = File Scan 0x80 = Modify 0x100 = Supervisor
rights_string	nvarchar(8)	varchar(8)	See srs.ncp_rights_string

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

### 3.2.6 SRS.CURRENT\_NTFS\_ACES

*Table 3-22 Current NTFS ACEs View*

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
fullpath	nvarchar(max)	text	Full UNC path to the file system entry
trustee_identity_system	nvarchar(256)	varchar(256)	Trustee's Identity System name
trustee_domain	nvarchar(256)	varchar(256)	Trustee's Active Directory domain
trustee_name	nvarchar(256)	varchar(256)	SAMAccount name
trustee_fdn	nvarchar(512)	varchar(512)	Full distinguished name
trustee_display_name	nvarchar(max)	text	<i>DOMAIN\SAMAccount</i>
trustee_type	integer	integer	0 = Unknown / Other 1 = User 2 = Group 3 = Computer 4 = Container 5 = Built-in domain 6 = Organization 7 = Organizational Unit 8 = Locality 9 = Domain DNS 10 = Volume 13 = Inet Org Person 16 = Country 17 = Tree Root 18 = Dynamic Group 20 = Alias 21 = eDirectory Pseudo Security Principal
sid	varbinary(68)	bytea	

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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
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

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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

### 3.2.7 SRS.CURRENT\_PERMISSIONS\_SCANS

*Table 3-23 Current 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 <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
platform	smallint	smallint	0 = Unknown 1 = Windows 2 = Linux 3 = Open Enterprise Server
filesystem	smallint	smallint	0 = Unknown 1 = NTFS 3 = NSS 4 = TFS 5 = Ext3 6 = ReiserFS 10 = Ext2 11 = Ext4 12 = Xfs 13 = Ramdisk 14 = OES_VFS 16 = Linux_generic 17 = NFS
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

## 3.2.8 SRS.PREVIOUS\_FS\_SCANDATA

Table 3-24 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 <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
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 for Active Directory Common Name (CN) for eDirectory
owner_fdn	nvarchar(512)	varchar(512)	Full distinguished object name
owner_display_name	nvarchar(max)	text	For Active Directory – <i>DOMAIN\SamAccountName</i> For eDirectory – FDN
owner_id	varbinary(68)	bytea	SID for Active Directory GUID for eDirectory



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	

### 3.2.9 SRS.PREVIOUS\_FS\_SCANDATA\_AD

*Table 3-25 Previous File System Scan Data for Active Directory View*

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
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			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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
size	bigint	bigint	For files, actual size; for directories, accumulative size of all subordinate files
size_on_disk	bigint	bigint	Assumes typical allocation unit size of 4K
size_compressed	bigint	bigint	Only accurate for NTFS file systems
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	<i>DOMAIN\SAMAccount</i>
owner_id	varbinary(68)	bytea	SID
ad_object_id	integer	integer	Reference to ad_objects table
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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
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	

### 3.2.10 SRS.PREVIOUS\_FS\_SCANDATA\_EDIR

*Table 3-26 Previous File System Scan Data for eDirectory View*

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
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

Column Name	SQL Server Data Type	PostgreSQL Data Type	Notes
size	bigint	bigint	For files, actual size; for directories, accumulative size of all subordinate files
size_on_disk	bigint	bigint	Assumes typical allocation unit size of 4K
size_compressed	bigint	bigint	Only accurate for NTFS file systems
owner_name	nvarchar(256)	varchar(256)	Common Name (CN)
owner_fdn	nvarchar(512)	varchar(512)	Full distinguished object name
owner_display_name	nvarchar(max)	text	Full distinguished object name
owner_id	varbinary(68)	bytea	GUID
edir_object_id	integer	integer	Reference to edir_objects table
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	

### 3.2.11 SRS.PREVIOUS\_FS\_SCANS

*Table 3-27 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 <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
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 2 = Linux 3 = Open Enterprise Server

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
filesystem	integer	integer	0 = Unknown 1 = NTFS 3 = NSS 4 = TFS 5 = Ext3 6 = ReiserFS 10 = Ext2 11 = Ext4 12 = Xfs 13 = Ramdisk 14 = OES_VFS 16 = Linux_generic 16 = Linux_generic 17 = NFS
scan_type	integer	integer	Should always be 1
progress_status	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending 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

## 3.2.12 SRS.PREVIOUS\_NCP\_TRUSTEES

Table 3-28 Previous NCP Trustees View

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
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_name	nvarchar(256)	varchar(256)	Common Name (CN)
trustee_fdn	nvarchar(512)	varchar(512)	Trustee's full distinguished name
trustee_display_name	nvarchar(512)	varchar(512)	Trustee's full distinguished name
trustee_type	integer	integer	0 = Unknown / Other 1 = User 2 = Group 3 = Computer 4 = Container 5 = Built-in domain 6 = Organization 7 = Organizational Unit 8 = Locality 9 = Domain DNS 10 = Volume 13 = Inet Org Person 16 = Country 17 = Tree Root 18 = Dynamic Group 20 = Alias 21 = eDirectory Pseudo Security Principal
trustee_guid	varbinary(68)	bytea	GUID
rights	integer	integer	0x01 = Read 0x02 = Write 0x04 = [undefined] 0x08 = Create 0x10 = Delete 0x20 = Access Control 0x40 = File Scan 0x80 = Modify 0x100 = Supervisor
rights_string	nvarchar(8)	varchar(8)	See srs.ncp_rights_string



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

### 3.2.13 SRS.PREVIOUS\_NTFS\_ACES

*Table 3-29 Previous NTFS ACEs View*

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
fullpath	nvarchar(max)	text	Full UNC path to the file system entry
trustee_identity_system	nvarchar(256)	varchar(256)	Trustee's Identity System name
trustee_domain	nvarchar(256)	varchar(256)	Trustee's Active Directory domain
trustee_name	nvarchar(256)	varchar(256)	SAMAccount name
trustee_fdn	nvarchar(512)	varchar(512)	Full distinguished name
trustee_display_name	nvarchar(max)	text	<i>DOMAIN\SAMAccount</i>
trustee_type	integer	integer	0 = Unknown / Other 1 = User 2 = Group 3 = Computer 4 = Container 5 = Built-in domain 6 = Organization 7 = Organizational Unit 8 = Locality 9 = Domain DNS 10 = Volume 13 = Inet Org Person 16 = Country 17 = Tree Root 18 = Dynamic Group 20 = Alias 21 = eDirectory Pseudo Security Principal
sid	varbinary(68)	bytea	

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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
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

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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

### 3.2.14 SRS.PREVIOUS\_PERMISSIONS\_SCANS

*Table 3-30 Previous 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 <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
platform	smallint	smallint	0 = Unknown 1 = Windows 2 = Linux 3 = Open Enterprise Server
filesystem	smallint	smallint	0 = Unknown 1 = NTFS 3 = NSS 4 = TFS 5 = Ext3 6 = ReiserFS 10 = Ext2 11 = Ext4 12 = Xfs 13 = Ramdisk 14 = OES_VFS 16 = Linux_generic 17 = NFS
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

### 3.2.15 SRS.BASELINE\_FS\_SCANDATA

Table 3-31 Baseline File System Scan Data Unified View

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
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 for Active Directory Common Name (CN) for eDirectory
owner_fdn	nvarchar(512)	varchar(512)	Full distinguished object name
owner_display_name	nvarchar(max)	text	For Active Directory – <i>DOMAIN\SamAccountName</i> For eDirectory – FDN
owner_id	varbinary(68)	bytea	SID for Active Directory GUID for eDirectory

Column Name	SQL Server Data Type	PosgreSQL 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	PosgreSQL 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	

### 3.2.16 SRS.BASELINE\_FS\_SCANDATA\_AD

*Table 3-32 Baseline File System Scan Data for Active Directory View*

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
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			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



Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
size	bigint	bigint	For files, actual size; for directories, accumulative size of all subordinate files
size_on_disk	bigint	bigint	Assumes typical allocation unit size of 4K
size_compressed	bigint	bigint	Only accurate for NTFS file systems
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	<i>DOMAIN\SAMAccount</i>
owner_id	varbinary(68)	bytea	SID
ad_object_id	integer	integer	Reference to ad_objects table
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

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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	

### 3.2.17 SRS.BASELINE\_FS\_SCANDATA\_EDIR

*Table 3-33 Baseline File System Scan Data for eDirectory View*

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
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

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
size	bigint	bigint	For files, actual size; for directories, accumulative size of all subordinate files
size_on_disk	bigint	bigint	Assumes typical allocation unit size of 4K
size_compressed	bigint	bigint	Only accurate for NTFS file systems
owner_name	nvarchar(256)	varchar(256)	Common Name (CN)
owner_fdn	nvarchar(512)	varchar(512)	Full distinguished object name
owner_display_name	nvarchar(max)	text	Full distinguished object name
owner_id	varbinary(68)	bytea	GUID
edir_object_id	integer	integer	Reference to edir_objects table
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	PosgreSQL 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	

### 3.2.18 SRS.BASELINE\_FS\_SCANS

*Table 3-34 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 <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
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 2 = Linux 3 = Open Enterprise Server

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
filesystem	integer	integer	0 = Unknown 1 = NTFS 3 = NSS 4 = TFS 5 = Ext3 6 = ReiserFS 10 = Ext2 11 = Ext4 12 = Xfs 13 = Ramdisk 14 = OES_VFS 16 = Linux_generic 17 = NFS
scan_type	integer	integer	Should always be 1
progress_status	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending 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

## 3.2.19 SRS.BASELINE\_NCP\_TRUSTEES

Table 3-35 Baseline NCP Trustees View

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
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_name	nvarchar(256)	varchar(256)	Common Name (CN)
trustee_fdn	nvarchar(512)	varchar(512)	Trustee's full distinguished name
trustee_display_name	nvarchar(512)	varchar(512)	Trustee's full distinguished name
trustee_type	integer	integer	0 = Unknown / Other 1 = User 2 = Group 3 = Computer 4 = Container 5 = Built-in domain 6 = Organization 7 = Organizational Unit 8 = Locality 9 = Domain DNS 10 = Volume 13 = Inet Org Person 16 = Country 17 = Tree Root 18 = Dynamic Group 20 = Alias 21 = eDirectory Pseudo Security Principal
trustee_guid	varbinary(68)	bytea	GUID
rights	integer	integer	0x01 = Read 0x02 = Write 0x04 = [undefined] 0x08 = Create 0x10 = Delete 0x20 = Access Control 0x40 = File Scan 0x80 = Modify 0x100 = Supervisor
rights_string	nvarchar(8)	varchar(8)	See srs.ncp_rights_string

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

### 3.2.20 SRS.BASELINE\_NTFS\_ACES

*Table 3-36 Baseline NTFS ACEs View*

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
identity_system	nvarchar(256)	varchar(256)	Identity system name <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
fullpath	nvarchar(max)	text	Full UNC path to the file system entry
trustee_identity_system	nvarchar(256)	varchar(256)	Trustee's Identity System name
trustee_domain	nvarchar(256)	varchar(256)	Trustee's Active Directory domain
trustee_name	nvarchar(256)	varchar(256)	SAMAccount name
trustee_fdn	nvarchar(512)	varchar(512)	Full distinguished name
trustee_display_name	nvarchar(max)	text	<i>DOMAIN\SAMAccount</i>
trustee_type	integer	integer	0 = Unknown / Other 1 = User 2 = Group 3 = Computer 4 = Container 5 = Built-in domain 6 = Organization 7 = Organizational Unit 8 = Locality 9 = Domain DNS 10 = Volume 13 = Inet Org Person 16 = Country 17 = Tree Root 18 = Dynamic Group 20 = Alias 21 = eDirectory Pseudo Security Principal
sid	varbinary(68)	bytea	



Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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
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

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
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

### 3.2.21 SRS.BASELINE\_PERMISSIONS\_SCANS

*Table 3-37 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 <ul style="list-style-type: none"> <li>◆ Active Directory – Forest FDN</li> <li>◆ eDirectory – Tree name</li> </ul>
domain	nvarchar(256)	varchar(256)	Active Directory domain
server	nvarchar(256)	varchar(256)	Server name

Column Name	SQL Server Data Type	PosgreSQL Data Type	Notes
scan_target	nvarchar(256)	varchar(256)	UNC root path for scan target
platform	smallint	smallint	0 = Unknown 1 = Windows 2 = Linux 3 = Open Enterprise Server
filesystem	smallint	smallint	0 = Unknown 1 = NTFS 3 = NSS 4 = TFS 5 = Ext3 6 = ReiserFS 10 = Ext2 11 = Ext4 12 = Xfs 13 = Ramdisk 14 = OES_VFS 16 = Linux_generic 17 = NFS
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

- ♦ [Section 3.3.1, “SRS.ACCESS\\_MASK\\_BASIC\\_STRING,” on page 84](#)
- ♦ [Section 3.3.2, “SRS.ACCESS\\_MASK\\_STRING,” on page 86](#)

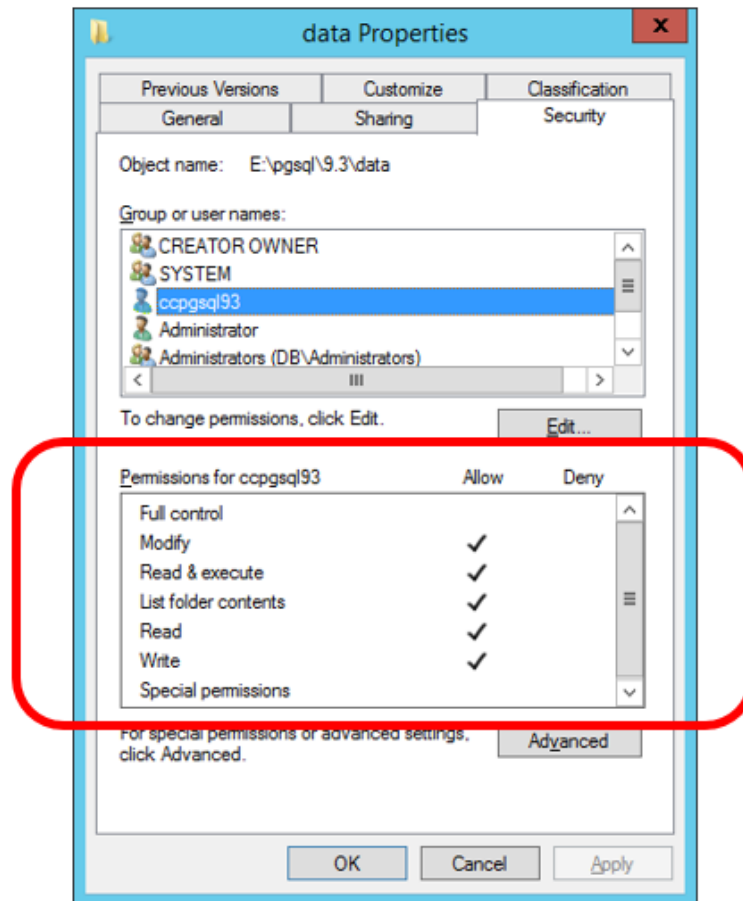
- ♦ Section 3.3.3, “SRS.AD\_ACCOUNT\_NAME,” on page 87
- ♦ Section 3.3.4, “SRS.ACE\_FLAGS\_STRING,” on page 88
- ♦ Section 3.3.5, “SRS.ACE\_TYPE\_STRING,” on page 89
- ♦ Section 3.3.6, “SRS.ATTRIBUTE\_STRING,” on page 90
- ♦ Section 3.3.7, “SRS.BYTE\_STRING,” on page 91
- ♦ Section 3.3.8, “SRS.BYTE\_UNIT\_STRING,” on page 91
- ♦ Section 3.3.9, “SRS.GUID\_BYTES,” on page 92
- ♦ Section 3.3.10, “SRS.GUID\_TEXT,” on page 92
- ♦ Section 3.3.11, “SRS.NCP\_RIGHTS\_STRING,” on page 93
- ♦ Section 3.3.12, “SRS.PATH\_HASH,” on page 94
- ♦ Section 3.3.13, “SRS.SID\_BYTES,” on page 94
- ♦ Section 3.3.14, “SRS.SID\_TEXT,” on page 95

### 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 and 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;

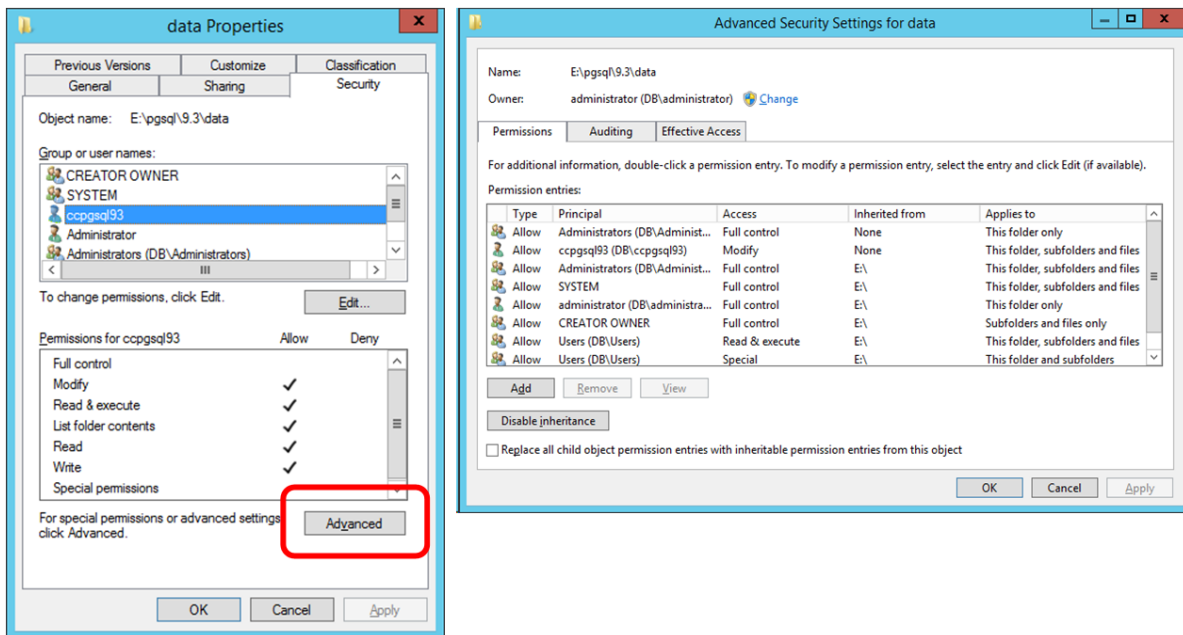
```

### 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('', '', 0x01020000000000052000000020020000);
```

### 3.3.4 SRS.ACE\_FLAGS\_STRING

Parameters	SQL Server	PostgreSQL
@flags	integer	integer
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	OI 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
0x00008000	Ex NSS Execute only
0x00010000	Sh NSS Shareable
0x00020000	Ds NSS Don't Sub-allocate
0x00040000	Tr NSS Transactional
0x00080000	In NSS Indexed
0x00100000	Ra NSS Read Audit
0x00200000	Wa NSS Write Audit
0x00400000	Ip NSS Immediate Purge
0x00800000	Ri NSS Rename Inhibit
0x01000000	Di NSS Delete Inhibit
0x02000000	Ci NSS Copy Inhibit
0x04000000	Dm NSS Don't Migrate

0x08000000	Ic	NSS Immediate Compress
0x10000000	Dc	NSS Don't Compress
0x20000000	Cc	NSS Can't Compress
0x40000000	Fa	NSS File Audit
0x80000000	Li	NSS Is a Link

### 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.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}')
```

### 3.3.10 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;
```

### 3.3.11 SRS.NCP\_RIGHTS\_STRING

Parameters	SQL Server	PostgreSQL
@rights	integer	integer
Return Value	nvarchar(8)	varchar(8)

**Description:** Converts an NCP trustee rights flag to its equivalent string representation. Flags correspond to the following values

0x0001	R	Read
0x0002	W	Write
0x0004		[Obsolete]
0x0008	C	Create
0x0010	E	Erase
0x0020	A	Access Control
0x0040	F	File Scan
0x0080	M	Modify
0x0100	S	Supervisor

Rights are displayed in the order: SRWCEMFA

## Example (SQL Server)

```
SELECT TOP(100)
    sd.fullpath,
    eo.fdn,
    srs.ncp_rights_string(ncp.rights) AS rights
FROM srs.ncp_trustees AS ncp
JOIN srs.scan_data AS sd ON ncp.scan_data_id = sd.id
JOIN srs.edir_objects AS eo ON ncp.trustee_guid = eo.guid;
```

## Example (PostgreSQL)

```
SELECT
    sd.fullpath,
    eo.fdn,
    srs.ncp_rights_string(ncp.rights) AS rights
FROM srs.ncp_trustees AS ncp
JOIN srs.scan_data AS sd ON ncp.scan_data_id = sd.id
JOIN srs.edir_objects AS eo ON ncp.trustee_guid = eo.guid
LIMIT 100;
```

### 3.3.12 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');
```

### 3.3.13 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;
```

### 3.3.14 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
```





# A

## Documentation Updates

This section contains information about documentation content changes that were made in this *Micro Focus File Reporter 3.0 Database Schema and Custom Queries Guide* after the initial release of File Reporter 2.5. The changes are listed according to the date they were published.

The documentation for this product is provided on the Web in two formats: HTML and PDF. The HTML and PDF documentation are both kept up-to-date with the changes listed in this section.

If you need to know whether a copy of the PDF documentation that you are using is the most recent, the PDF document includes a publication date on the title page.

The documentation was updated on the following dates:

### A.1 July 2, 2018

Updates were made to the following sections:

Location	Update Description
<a href="#">Table 1-1 on page 8.</a>	Updated table.
<a href="#">"Filter by File Name" on page 22.</a>	Updated reference for PostgreSQL.
<a href="#">Section 3.1.1, "ANALYSIS.FILE_SCAN_ENTRIES," on page 25.</a>	New table.

### A.2 July 19, 2016

Updates were made to the following sections:

Location	Update Description
<a href="#">Section 1.1, "Updates and Breaking Changes," on page 7.</a>	Updates detailed.
<a href="#">Section 3.3, "Functions," on page 83</a>	New functions detailed.

### A.3 April 27, 2015

Updates were made to the following sections:

Location	Update Description
Various locations	Previous references to "active," have been changed to "current."

Location	Update Description
Section 1.1, "Updates and Breaking Changes," on page 7.	New section.
Section 1.5, "Supported Database Views," on page 9.	Updated with various Previous and Baseline views.
Section 3.1.10, "SRS.SCANS," on page 31.	<p>Values for <code>progress_status</code> have been updated with new values.</p> <p>The definition notes for <code>is_valid</code> have been updated to [Deprecated].</p>
Section 3.2, "Views," on page 41.	<p>Definitions have been added for new Previous and Baseline views.</p> <p>Notes and definitions have been added in a Notes column.</p>
Section 3.2.4, "SRS.CURRENT_FS_SCANS," on page 48.	<p>Values for <code>progress_status</code> have been updated with new values.</p> <p>The definition notes for <code>is_valid</code> have been updated to [Deprecated].</p>
Section 3.2.7, "SRS.CURRENT_PERMISSIONS_SCANS," on page 54.	<p>Values for <code>progress_status</code> have been updated with new values.</p> <p>The definition notes for <code>is_valid</code> have been updated to [Deprecated].</p>
Various locations	The designation of <code>int</code> has been replaced with <code>integer</code> .