

# SpaceObServer – Database structure (V7)

## Database Tables

### Table ROOTS:

This table is the foundation in each SpaceObServer database. It holds all “root”-directories that were chosen to be scanned. Additional information like when to start the next scan or which scan options to use, are also stored here.

#### Table fields:

ROOT_ID	Unique identifier for the root.
PATH	The full path of the root.
FIRSTSCAN_TIME	The date when the root was scanned first.
LASTSCAN_TIME	The date when the root was scanned last.
CAPACITY	The capacity of the hard disk where the root is located.
FREESPACE	The free space of the hard disk where the root is located.
CLUSTER_SIZE	The cluster size of the hard disk where the root is located.
SCANINTERVALL	Indicates the interval in which the root is scanned.
SCANNOW	A flag that indicates if the user manually ordered to start a scan immediately.
STATUS	Used within a scan (Holds the currently scanning directory path of the root)
SCANHOUR	The daytime when the scheduled scan of the root should be executed.
NEXTSCAN_TIME	The time stamp when the next time a scan should be executed.
LAST_DURATION	The duration of the last scan.
ERROR	Used within a scan (Holds the last error occurred within a scan)
FOLLOWREPARSEPOINTS	Indicates if mount points and symbolic links within the file system to scan should be followed. “Scan > Configure Scans > Scan Options > Follow Mount Points and External Symbolic Links”
SCAN_PRIORITY	Tells the system service the thread priority to run scans for the root.
HISTORY_PERIOD	Holds the interval for how long historic file system information should be hold in the database. “Scan > Configure Scans > Additional Options > Purge Data

	older then”
ERROR_CODE	Stored the last error code that occurs when scanning the root.
NEXT_USN	Used Only for NTFS Change Journal analysing. (Holds the unique identifier of the last analysed change journal entry)
DATA_TABLE_PREFIX	Indicates in which tables the file system information of this root are stored in. See (1)
FILES_SCANNED	<Not official used>
INCLUDE_ADS	Indicates if Alternate Data Streams and Hardlinks should be tracked within scans of the root. “Scan > Configure Scans > Scan Options > Track NTFS Alternate Data Streams and NTFS Hardlinks”
USE_CHGJRL	Indicates if the root should update sizes in the database by analysing NTFS Change Journals.
SERVER_NAME	The name of the observing server for the root.
DATA_BASE	The database to store the scan information for this root in. “Scan > Configure Scans > Expert Settings > Database”.
TOTAL_FILES	The total number of files located in the root.
SCANNED_FILES	Used within a scan. (The number of currently scanned files)
ONLY_INIT_SCANS	Execute each update scan as new initial scan, using “fresh” database tables “Scan > Configure Scans > Additional Options > Purge Data older then”
SSH_SERVER_NAME	For SSH scans only: The name/IP of the server to scan.
SSH_SERVER_PORT	For SSH scans only: The port used on this server for SSH.
SSH_USER_NAME	Reserved field (no longer used)
SSH_USER_PWD	Reserved field (no longer used)
CLEAN_COLLECT	Used within a scan. (Indicates if the next scan of this root is a “clean collect”, that means: Even information of files that have not been “touched” since the last scan will be updated in the database)
SAVE_MD5_CHECKSUM	Indicates if for the files of this root the “MD5 check sum” of the file content should be calculated. “Scan > Configure Scans > Scan Options > Calculate MD5 checksums for files”

REFRESH_ROOTS	If this field is true, the SpaceObServer Agent, responsible for this root, will refresh its root settings.
FULL_PATH_SET	True if all fields, "FULL_PATH" of entries in the corresponding X_FOLDERS table filled with information.
SCHEDULE_WEEK_DAYS	A representation of the week days on which a scan for this root directory is scheduled (1=Sunday, 2=Monday, 4=Tuesday,... 3=Sunday and Monday, 5=Sunday and Tuesday,...).
SCHEDULE_MONTH_DAYS	A representation of the month days on which a scan for this root directory is scheduled (1=1 <sup>st</sup> , 2=2 <sup>nd</sup> , 4=3 <sup>rd</sup> ,... 3=1 <sup>st</sup> and 2 <sup>nd</sup> , 5=1 <sup>st</sup> and 3 <sup>rd</sup> ,...).
EXCLUDE_PATTERN	The patterns for files and directories to exclude when scanning (See: "Scan > Configure Scans > Scan Filter").
NOTIFICATION	Used internally to notify the service that some kind of operation (Scanning, MD5 checksum calculation,...) has to be executed for this root directory.
PARENT_ID	The ID of the "Container" where this root is included in.
FLAGS	Preserved for internal usage only.
ROLE_READ_DATA	A list of users or groups which are allowed to see this root/scan data in the SpaceObServer user interface (See "Scan > Configure Scans > Expert Settings > User Access").
COST_VALUE	Used to calculate the "Cost" of this roots files and directories (can be shown in the "Details" list and all exports).
COST_UNIT	The unit of the "Cost" of this roots files and directories (See also "COST_VALUE")
DIR_TREE_SIM	Indicates if Similar Folders statistic should be calculated for this root. "Scan > Configure Scans > Scan Options > Calculate Similar Folder statistics".
SAVE_PERMISSIONS	Indicates if Permissions should be queried and stored in the database for this root. "Scan > Configure Scans > Scan Options > Calculate Permissions and store in database".
APP_VERSION	The version of SpaceObServer, which executed the last scan of this root.
USER_NAME	The login name for scan authentication configured at "Scan > Configure Scans > Expert Settings > Use different credentials for scan".
PASSWORD	The login password (encrypted) for scan authentication configured at "Scan > Configure Scans > Expert Settings > Use different credentials for scan".

(1) DATA\_TABLE\_PREFIX:

File system information of different ROOTS are normally stored in different database tables. Normally for each new entry in the table ROOTS a new database table structure composed of four tables are created. These new Tables are named:

< DATA\_TABLE\_PREFIX >\_FOLDERS

< DATA\_TABLE\_PREFIX >\_FILES

< DATA\_TABLE\_PREFIX >\_FolderVersion

< DATA\_TABLE\_PREFIX >\_FileVersion

These four tables hold all the information needed to produce the SpaceObServer reports.

In the following these four tables are introduced.

**Table X\_FOLDERS:**

This table holds information about the scanned directory structure. It also stores size, date and other information for each folder. The hierarchical folder structure is mapped to the flat database tables using [nested sets](#).

Table fields:

FO_ID	Unique identifier for the folder.
PARENT_ID	The FO_ID of the parent folder.
NS_LEFT	The lower nested set border.
NS_RIGHT	The upper nested set border.
CURRENT_NAME	The name of the folder.
CREATION_TIME	Creation date of the folder.
DELETION_TIME	Deletion date of the folder.
C_REAL_SIZE	The current size of the folder (in bytes).
C_ALLOCATED_SIZE	The current allocated space of the folder (in bytes).
C_LASTACCESS_TIME	The current last access date of the folder.
C_LASTWRITE_TIME	The current last change date of the folder.
C_FILE_COUNT	The current number of files located in this folder.
C_OWNER	The current file owner.
C_ATTRIBUTES	The current file attributes.
ERROR_CODE	Stored the last error code that occurs when scanning this directory.
FRN	NTFS File Reference Number of this folder. (Needed for analyzing NTFS Change Journals to update changes in the database)
ANNOTATION	Holds an annotation for this folder. (The annotation can be set by user in the SpaceObServer user interface)
FLAGS	Here are some states of this folder stored, which are only used within a scan.
FULL_PATH	This is the full path of this directory.
CHECK_STATE	Holds the check-mark state of the directory, which was stored from the user interface to the database ("Save Check States to Database" from the context menu of the SpaceObServer directory tree).
PERMISSIONS	Reference to the permissions list (ACL) of this folder.

**Table X\_FILES:**

This table holds information about each scanned file. It stores size, date and other information for each file.

Table fields:

FI_ID	Unique identifier for the file.
CURRENT_NAME	The name of the file.
PARENT_ID	The FO_ID of the parent folder.
PARENT_NSLEFT	The NS_LEFT value of the parent folder.
CREATION_TIME	Creation date of the file.
DELETION_TIME	Deletion date of the file.
DETECTEDWHILESCAN	A flag used only within update scans. (Indicates if an existing folder was detected within the last update scan. If not, the folder was removed)
C_REAL_SIZE	The current size of the file (in bytes).
C_ALLOCATED_SIZE	The current allocated space of the file (in bytes).
C_LASTACCESS_TIME	The current last access date of the file.
C_LASTWRITE_TIME	The current last change date of the file.
C_ATTRIBUTES	The current file attributes.
C_FILETYP	The current file extension.
C_OWNER	The current file owner.
ANNOTATION	Holds an annotation for this file. (The annotation can be set by user in the SpaceObServer user interface)
C_HARDLINKS	The number of NTFS Hardlinks on the file.
CHECK_STATE	Holds the check-mark state of the file, which was stored from the user interface to the database ("Save Check States to Database" from the context menu of the SpaceObServer directory tree).
PERMISSIONS	Reference to the permissions list (ACL) of this file.
MD5_CHECKSUM_LOW	The first 64 bit of the MD5 checksum for this file. (This field and MD5_CHECKSUM_HIGH are optional and only created if the MD5 checksum was activated for this root).
MD5_CHECKSUM_HIGH	The second 64 bit of the MD5 checksum for this file.

Note: C\_REAL\_SIZE, C\_ALLOCATED\_SIZE, C\_WASTED\_SPACE, C\_LASTACCESS\_TIME, C\_LASTWRITE\_TIME, C\_ATTRIBUTES, C\_FILE\_COUNT, C\_FILETYP and C\_OWNER are cached/redundant values.

All these fields in the "FILES" table have a corresponding field in the "FileVersion"

table (field name without "C\_"). The values in these fields are equal to the values of the latest "FileVersion" entry for the corresponding file.  
The fields C\_ATTRIBUTES and C\_OWNER in the "FOLDERS" table have a corresponding field in the "FolderVersion" table. The other fields are merged values of all files located in this folder.

**Table X\_FolderVersion:**

This table archives historic information folders, even when they were deleted or renamed. With the help of the content of this table, SpaceObServer can show former file system structures.

Table fields:

ID	Unique identifier for this entry.
FO_ID	Reference to the corresponding folder in the FOLDERS table.
NAME	The name of the folder at the time when this version was up-to-date.
ATTRIBUTES	The attributes of the folder at the time when this version was up-to-date.
OWNER	The owner of the folder at the time when this version was up-to-date.

**Table X\_FileVersion:**

This table archives historic file sizes, even when they were deleted or renamed. With the help of the content of this table, SpaceObServer can show the size development of file systems.

Table fields:

ID	Unique identifier of the entry.
FI_ID	Reference to the corresponding file in the FILES table.
REAL_SIZE	The size of the file at the time when this version was up-to-date.
ALLOCATED_SIZE	The allocated space of the file at the time when this version was up-to-date.
LASTACCESS_TIME	The last access date of the file at the time when this version was up-to-date.
LASTWRITE_TIME	The last change date of the file at the time when this version was up-to-date.
ATTRIBUTES	The attributes of the file at the time when this version was up-to-date.
OWNER	The owner of the file at the time when this version was up-to-date.



## Indexes

Find all used indexes, and the reason why they are used, in the table "Indexstruktur des SpaceObServers.xls".

## Date time values

The representation of dates in the SpaceObServer database is equal to the OLE/Com date format. We use an equal format for each type of database to make it easier to support different SQL databases.

The integral part of a date value is the number of days that have passed since 12/30/1899. The fractional part of the value is fraction of a 24-hour day that has elapsed. Following are some examples of OLE/Com date values and their corresponding dates and times:

0	12/30/1899 12:00 am
2.75	1/1/1900 6:00 pm
-1.25	12/29/1899 6:00 am
35065	1/1/1996 12:00 am

To find the fractional number of days between two dates, simply subtract the two values, unless one of the values is negative. Similarly, to increment a date and time value by a certain fractional number of days, add the fractional number to the date and time value if the value is positive. When working with negative values, computations must handle time portion separately. The fractional part reflects the fraction of a 24-hour day without regard to the sign of the value. For example, 6:00 am on 12/29/1899 is  $-1.25$ , not  $-1 + 0.25$ , which would be  $-0.75$ . There are no values between  $-1$  and  $0$ .

To convert this value to the MS SQL Server data type datetime us the DATEADD() function:

```
SELECT DATEADD(dd , LASTSCAN_TIME, '18991230') FROM ROOTS --Days only
```

```
SELECT DATEADD(hh , LASTSCAN_TIME*24, '18991230') FROM ROOTS --with hours
```

```
SELECT DATEADD(mi , LASTSCAN_TIME*24*60, '18991230') FROM ROOTS --with min.
```

## File Attributes

There is a field "ATTRIBUTES" in the table X\_FileVersion / X\_FolderVersion ("C\_ATTRUBUTES" in X\_FILES/X\_FOLDERS). The value represents the file attributes (Temporary, System, ...). Each of these attributes has a specific value

(See: Attribute values below). The value in this field is equal to the sum of all attributes of the file.

**Attribute values (Hexadecimal)**

FILE_ATTRIBUTE_READONLY	\$00000001
FILE_ATTRIBUTE_HIDDEN	\$00000002
FILE_ATTRIBUTE_SYSTEM	\$00000004
FILE_ATTRIBUTE_DIRECTORY	\$00000010
FILE_ATTRIBUTE_ARCHIVE	\$00000020
FILE_ATTRIBUTE_NORMAL	\$00000080
FILE_ATTRIBUTE_TEMPORARY	\$00000100
FILE_ATTRIBUTE_COMPRESSED	\$00000800
FILE_ATTRIBUTE_OFFLINE	\$00001000
FILE_ATTRIBUTE_ADS ("Alternative Data Stream" included)	\$00002000