unicon

Scout Database Model for CR versions

White Paper

Date: 2021-11-16

0. Legal information

Unicon Software Entwicklungs- und Vertriebsgesellschaft mbH
Ludwig-Erhard-Allee 26
76131 Karlsruhe
+49 (0) 721 96451-0

1. About the Scout Enterprise Database Model (CR)

The hardware-independent operating system eLux is a Linux-based desktop solution for Thin Clients and PCs. eLux provides fast, comfortable and secure access to Windows and other servers in a server-based environment - without any Linux know-how.

eLux clients, and even Windows clients, can easily be managed with the Scout Enterprise Management Suite. The administrator configures all devices in organizational units from the Scout Console. Of course, he also can define the required applications, transfer files and perform firmware updates remotely. For troubleshooting, there are provided diagnostic tools and mirroring. If you want to equip some administrators with particular remote functions, you can alternatively use the web-based management console for them, Scout Enterprise Dashboard.

The Scout Enterprise solution provides database support and is especially designed to manage large-scale installations. More details and the latest versions of Scout Enterprise are available for download on www.myelux.com.

In the following, the data base model for the current CR version of the Scout Enterprise Management Suite is described. For the LTSR version, see the relevant white paper.

For a graphical overview see the PDF of the Scout Enterprise database model.

The tables are grouped into the following sections

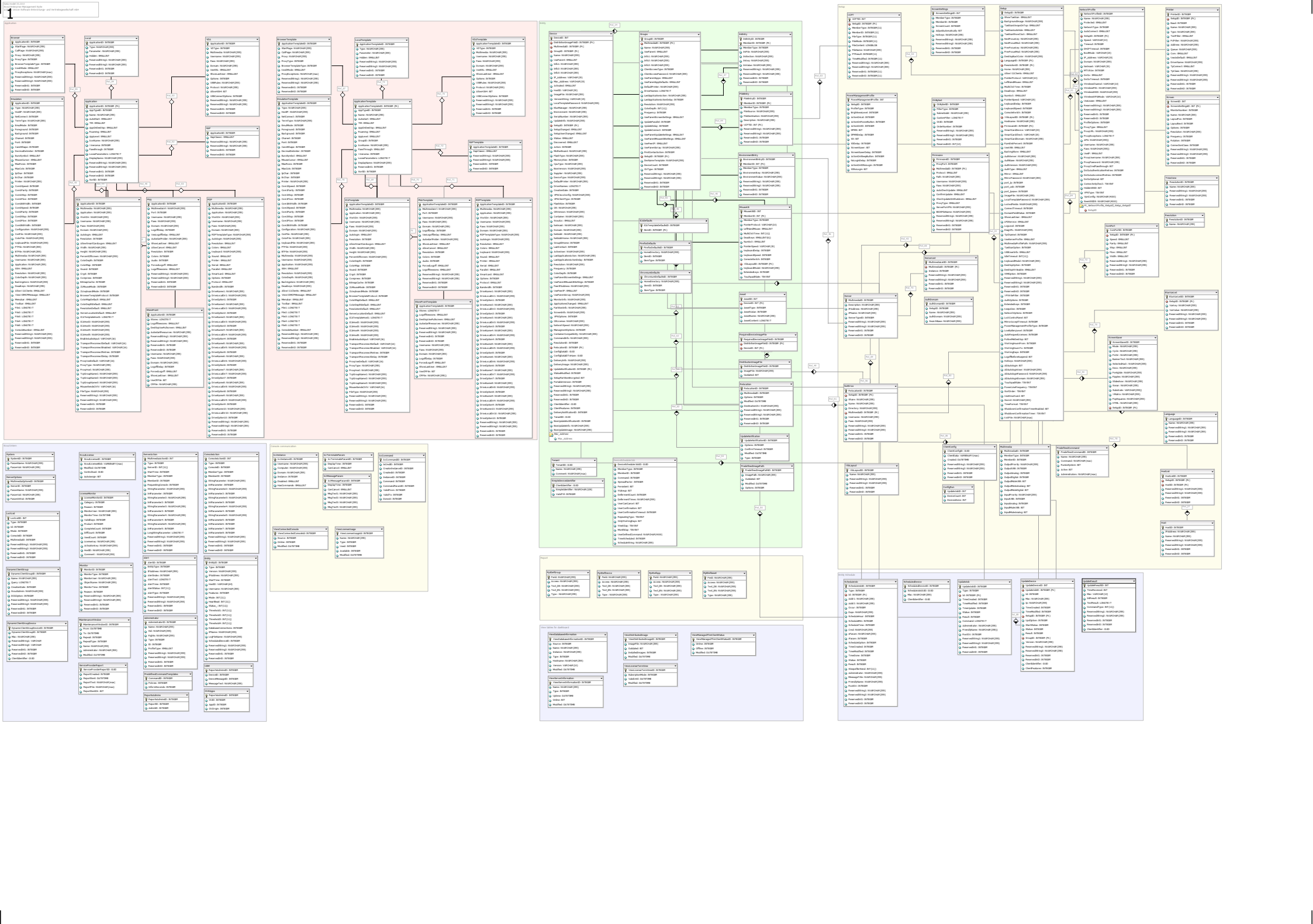
- **Entity** contains tables of the hierarchical structure of OUs/groups and devices
- **Application** contains tables for application definitions
- **Setup** contains tables with device configuration data on all levels (OUs/groups and devices)
- **Entity Schedule** contains tables storing data of the Scout scheduler mechanism
- **Scout Enterprise intern** groups a set of internal tables used by Scout Server and Consoles to interact

In case of any problems or questions it is highly recommended that you enter data by using the Scout Console and subsequently view the database tables.

Note
This White Paper refers to Scout Enterprise Database Model version 15.2110 CR.

Date: 2022-10-14
Project: Enterprise Management Suite
Copyright: Frank Steffen Informatik-Dienst- und Vertriebsgesellschaft mbH

1



3. Entity

3.1. Table: Assets

This table holds all asset values the client sends to the Scout Enterprise manager.

ColumnName	DataType	Key	Not Null	Default Value	Comment
AssetID	INTEGER	PK	NN		Unique file entry ID
DeviceID	INTEGER				ID of the device the asset entry belongs to
AssetType	INTEGER				Type for the asset
AssetValue	NVARCHAR (255)				value for the asset
AssetName	NVARCHAR (255)				Name of the asset
AssetContent	NVARCHAR (max)				Content of the asset

IndexName	IndexType	Columns
PRIMARY	PRIMARY	AssetID

How to create

```
CREATE TABLE Asset(
AssetID INTEGER NOT NULL;
DeviceID INTEGER NOT NULL;
AssetType INTEGER NOT NULL;
AssetValue INTEGER NOT NULL;
AssetName NVARCHAR(255) NULL;
AssetContent NVARCHAR(max) NULL;
PRIMARY KEY (AssetID));
```

3.2. Table: ChromiumDefaults

This table stores the Chromium software defaults.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ChromiumDefaultsID	INTEGER IDENTITY (1,1)	PK	NN		Unique ID
HomeDirectory	NVARCHAR(255)				holds the home directory
ItemID	INTEGER		NN		OU ID
ItemType	INTEGER		NN		always 1 (TYPE_OU) at this time

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ChromiumDefaultsID

How to create

```
CREATE TABLE ChromiumDefaults(
ChromiumDefaultsID INTEGER IDENTITY (1,1) NOT NULL;
HomeDirectory NVARCHAR(255);
ItemID INTEGER NOT NULL;
ItemType INTEGER NOT NULL;
CONSTRAINT PK_ChromiumDefaults;
PRIMARY KEY (ChromiumDefaultsID));
```

3.3. Table: Device

Each entry represents a device which is member of a group. Many attributes are marked to be set by the client or by the Scout Enterprise server. These values should not be changed since they will be overwritten.

Column name	Data type	Key	Not Null	Default Value	Comment
DeviceID	INTEGER	PK	NN		Unique device ID
WolServerID	INTEGER	FK			WakeOnLAN server ID (foreign key refers to table ServerID)
GroupID	INTEGER	FK	NN		ID of the owning group (foreign key refers to table GroupID)
Name	NVARCHAR(255)		NN		Unique name of device
UseParent	SMALLINT(6)		NN	1	Use setup of the owning group **0|1**
Info1	NVARCHAR(255)				Arbitrary user information
Info2	NVARCHAR(255)				Arbitrary user information
Info3	NVARCHAR(255)				Arbitrary user information
IP_Address	NVARCHAR(15)		NN		IP address (dotted notation)
Mac_Address	NVARCHAR(15)		NN		MAC address
Activated	SMALLINT(6)		NN		Management enabled **0|1**

Column name	Data type	Key	Not Null	Default Value	Comment
HostID	NVARCHAR(30)				Client host ID
ImageFile	NVARCHAR(255)				Currently installed IDF file (set by client)
VersionString	NVARCHAR(15)				Version (set by client)
LocalPassword	NVARCHAR(255)				Encrypted device password
EluxManager	NVARCHAR(255)				Manager IP address
Environment	NVARCHAR(255)				obsolete
Serial Number	NVARCHAR(255)				Serial number (set by client)
UpdateInfo	NVARCHAR(255)			-1,0,0,0,0	Update state (set by Scout Enterprise server)
CommandInfo	NVARCHAR(255)				
SetupID	INTEGER	FK		0	Setup ID (foreign key refers to table Setup.SetupID)
SetupChanged	SMALLINT(6)				obsolete
SetupUserChanged	SMALLINT(6)				set by Scout Enterprise server
Status	SMALLINT(6)			99	Device state (set by Scout Enterprise server) 0 Device is off or unreached 1 Device discovery in progress 2 Device logon in progress 3 Device up and running 4 Device updating 99 Device state unknown

Column name	Data type	Key	Not Null	Default Value	Comment
Discovered	SMALLINT(6)			0	set by Scout Enterprise server
Action	INTEGER			11	Action to take place if device contacts manager 0 ACTION_NOTHING Do nothing or a combination of the following values: 1 ACTION_SETUPCHANGED Device must refetch setup data 2 ACTION_USERSETUPCHANGED Device must refetch user setup data 4 ACTION_XRESTART Device must re-logon 8 ACTION_REBOOT Device must restart 16 ACTION_REFRESH Device must refresh setup data
Motherboard	NVARCHAR(255)				Motherboard description (set by client)
FlashType	NVARCHAR(255)				Flash type (set by client)
Memorysize	INTEGER				Memory size (set by client)
CpuType	NVARCHAR(255)				CPU type (set by client)
Biosversion	NVARCHAR(255)				BIOS version (set by client)
Supplier	NVARCHAR(255)				Device supplier (set by client)
DeviceType	NVARCHAR(255)				Device type (set by client)
DefaultPrinter	NVARCHAR(255)				Name of default printer (must exist in tablePrinter for the according SetupID)
DriverNames	NVARCHAR(max)				

Column name	Data type	Key	Not Null	Default Value	Comment
CreationDate	INTEGER				Entry creation time stamp
VPNCiscoConfig	NVARCHAR(255)				VPN configuration (Cisco, VPNC, L2TP)
VPNClientType	INTEGER			0	VPN client to be used **0** No VPN client **1** F-Secure **2** FreeS/WAN **3** Cisco **4** PPTP **5** VPNC **6** L2TP
FlashSize	INTEGER			0	Flash size (set by client)
OS	NVARCHAR(255)				Operating system (set by client)
OSVersion	NVARCHAR(255)				Version of operating system (set by client)
Container	NVARCHAR(255)				Container identification (set by client)
License BuiltIn ScoutLic	SMALLINT(6)				Management license (set by Scout Enterprise server)
Netmask	NVARCHAR(255)				Network mask (set by Scout Enterprise server)
Domain	NVARCHAR(255)				Domain name (set by client)
NetAddr	NVARCHAR(255)				Network address (set by client)
NetAddrHome	NVARCHAR(255)				Home network address (set by server)

Column name	Data type	Key	Not Null	Default Value	Comment
GroupIDHome	INTEGER			-1	Home group ID (set by server)
LastContact	INTEGER			0	Last contact time stamp
ActiveUser	NVARCHAR(255)				Last user (set by client)
LastApplicationAction	NVARCHAR(255)			#NONE	Action takes place when the last application was closed **#NONE** Do nothing **#RESTARTX** Logoff user **#REBOOT** Reboot device **#OFF** Shutdown the device **#LOCK** Lock the device **#VPNDISCON** Disconnect from VPN
LastApplicationAction Delay	INTEGER			0	Delay (in seconds) before LastApplicationAction is processed
Resolution	NVARCHAR(255)			1024x768	Screen resolution **640x480|800x600|1024x768|1152x864|1280x1024|1600x1200**
Frequency	INTEGER			60	Screen frequency **60|70|75|80|85|90|100**
ColorDepth	INTEGER			16	Screen color depth **8|16|24**
UseDDC	SMALLINT(6)			1	Use DDC settings of this device **0|1**

Column name	Data type	Key	Not Null	Default Value	Comment
UseParentScreen Settings	SMALLINT(6)			1	Use screen settings of the owning group 0|1
UseParentMouseKB Settings	SMALLINT(6)			1	Use mouse and keyboard settings of the owning group 0|1
PeerIPAddress	NVARCHAR(255)				Peer IP address (set by Scout Enterprise)
UsePeerIP	SMALLINT(6)			0	Use peer IP (e.g access point with NAT) when connecting the client (instead of device IP) 0|1
UseParentArray	NVARCHAR(255)			FF	Bit field indicates which property should be used from the owning group
MonitorInfo	NVARCHAR(255)				Semicolon-separated info about connected monitor info (set by client):<Model>;<Vendor>;<Serial#>
ApplicationsChanged	SMALLINT(6)			1	Indicates whether applications have changed and must be sent to the client 0|1
PartitionInfo	NVARCHAR(255)				Partition info (set by client)
ScreenInfo	NVARCHAR(255)				Semicolon-separated info about current screen resolution (set by client): <Width>;<Height>;<Depth>;<HFreq[kHz]>;<VFreq[Hz]>
VPNOptions	INTEGER			1	1 VPNOPT_AUTOCONNECT
OSLicense	NVARCHAR(255)				The license of the OS for Windows XPe or Windows CE devices
NetworkSpeed	NVARCHAR(255)				The actual network speed (set by client)
ManagementOptions	INTEGER				

Column name	Data type	Key	Not Null	Default Value	Comment
ContainerCompatibility	NVARCHAR(255)				Holds the comma-separated container compatibility list Example: UC_RP, UC_RL. That means the clients hardware is able to run either eluxRP or eluxRL.
CurrentMac	NVARCHAR(255)				If a client is switching to another MAC address (e.g. enabling wireless lan) this is the actual MAC address
OUCustomFilter	NVARCHAR(max)				The client is able to send a custom filter by which the scout server may decide to which OU the device should belong to.
DistributionImageFile	INTEGER				Reference to the DistributionImageFile table
TimezoneID	INTEGER				When the Scout management initiates a client relocation, this is the ID to the Relocation table
RelocationID	INTEGER				The last config snapshot ID the client has confirmed
ConfigDataID	UNIQUEIDENTIFIER				Reference to the `ClientConfig` table
ConfigDataIDToHave	UNIQUEIDENTIFIER				Reference to the `ClientConfig` table
DeliveryInfo	NVARCHAR				Stores the result values of the last delivery command
DeliveryImage	NVARCHAR(255)				Shows the last delivery image
UpdateNotificationID	INTEGER				New field to handle the 'Image update notification' feature (14.9)
DeliveryNotificationID	INTEGER				
BiosUpdateNotificationID	INTEGER				ID of an UEFI update notification
BiosUpdateInfo	NVARCHAR(255)				UEFI update result

unicon

Column name	Data type	Key	Not Null	Default Value	Comment
BiosUpdateImage	NVARCHAR(255)				UEFI file name the client sent back to the server
Note	NVARCHAR(255)				
StatusModified	INTEGER		NN	0	
SetupPartitionEncrypted	BIT				
PortableVersion	INTEGER				
ClientIdentifier	UNIQUEIDENTIFIER		NN		Unique identifier for client devices and eLux portable sticks
ClientFeatures	INTEGER		NN		0 CLIENT_FEATURE_NONE 1 LIENT_FEATURE_CLIENT_IDENTIFIER 2 CLIENT_FEATURE_BIOS_UPDATE
TenantID	UNIQUEIDENTIFIER				Unique identifier for tenants in SPM mode (assign tenants to devices)
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	DeviceID
Mac_address	Index	MAC_address

unicon

How to create

```
CREATE TABLE Device(
DeviceID INTEGER NOT NULL;
WolServerID INTEGER;
GroupID INTEGER NOT NULL;
Name NVARCHAR(255) NOT NULL;
UseParent SMALLINT NOT NULL;
Info1 NVARCHAR(255);
Info2 NVARCHAR(255);
Info3 NVARCHAR(255);
IP_Address NVARCHAR(15) NOT NULL;
Mac_Address NVARCHAR(15) NOT NULL;
Activated SMALLINT NOT NULL;
HostID NVARCHAR(30);
ImageFile NVARCHAR(255);
VersionString NVARCHAR(15);
LocalPassword NVARCHAR(255);
EluxManager NVARCHAR(255);
Environment NVARCHAR(255);
SerialNumber NVARCHAR(255);
UpdateInfo NVARCHAR(255);
CommandInfo NVARCHAR(255);
SetupID INTEGER;
SetupChanged SMALLINT;
SetupUserChanged SMALLINT;
Status SMALLINT;
Discovered SMALLINT;
Action INTEGER;
```

```
MotherBoard NVARCHAR(255);
FlashType NVARCHAR(255);
MemorySize INTEGER;
CpuType NVARCHAR(255);
BiosVersion NVARCHAR(255);
Supplier NVARCHAR(255);
DeviceType NVARCHAR(255);
DefaultPrinter NVARCHAR(255);
DriverNames NVARCHAR(max);
CreationDate INTEGER;
VPNCiscoConfig NVARCHAR(255);
VPNClientType INTEGER;
FlashSize INTEGER;
OS NVARCHAR(255);
OSVersion NVARCHAR(255);
Container NVARCHAR(255);
ScoutLic SMALLINT;
Netmask NVARCHAR(255);
Domain NVARCHAR(255);
NetAddr NVARCHAR(255);
NetAddrHome NVARCHAR(255);
GroupIDHome INTEGER;
LastContact INTEGER;
ActiveUser NVARCHAR(255);
LastApplicationAction NVARCHAR(255);
LastApplicationActionDelay INTEGER;
Resolution NVARCHAR(255);
Frequency INTEGER;
ColorDepth INTEGER;
```

```
UseParentScreenSettings SMALLINT;
PPTPPassword NVARCHAR(255);
UseParentMouseKBSettings SMALLINT;
PeerIPAddress NVARCHAR(255);
UsePeerIP SMALLINT;
UseParentArray NVARCHAR(255);
MonitorInfo NVARCHAR(255);
ApplicationsChanged SMALLINT;
PartitionInfo NVARCHAR(255);
ScreenInfo NVARCHAR(255);
NetworkSpeed NVARCHAR(255);
ManagementOptions INTEGER;
ContainerCompatibility NVARCHAR(255);
CurrentMac NVARCHAR(255);
OUCustomFilter NVARCHAR(max);
DistributionImageFileID INTEGER;
TimeZoneID INTEGER;
RelocationID INTEGER;
ConfigDataID UNIQUEIDENTIFIER;
ConfigDataIDToHave UNIQUEIDENTIFIER;
DeliveryInfo NVARCHAR(255);
DeliveryImage NVARCHAR(255);
UpdateNotificationID INTEGER;
BiosUpdateInfo __UNICON_VARCHAR__(255);
DeliveryNotificationID INTEGER;
BiosUpdateNotificationID INTEGER;
BiosUpdateImage NVARCHAR(255);
Note NVARCHAR(255);
StatusModified INTEGER NOT NULL;
```

```
SetupPartitionEncrypted BIT;
PortableVersion INTEGER;
ClientIdentifier UNIQUEIDENTIFIER NOT NULL CONSTRAINT DF_Device_ClientIdentifier DEFAULT NEWID();
ClientFeatures INTEGER NOT NULL CONSTRAINT DF_Device_ClientFeatures DEFAULT 0;
TenantID UNIQUEIDENTIFIER CONSTRAINT DF_Device_TenantID DEFAULT '00000000-0000-0000-0000-000000000000'
ReservedString1 NVARCHAR(255);
ReservedString2 NVARCHAR(255);
ReservedInt1 INTEGER;
ReservedInt2 INTEGER;
CONSTRAINT FK_Device_GroupID_Groups_GroupID FOREIGN KEY (GroupID) REFERENCES Groups (GroupID);
CONSTRAINT FK_Device_WolServerID_Server_ServerID FOREIGN KEY (WolServerID) REFERENCES Server (ServerID);
CONSTRAINT FK_Device_RelocationID FOREIGN KEY (RelocationID) REFERENCES Relocation (RelocationID);
CONSTRAINT FK_Device_SetupID_Setup_SetupID FOREIGN KEY (SetupID) REFERENCES Setup (SetupID);
CONSTRAINT FK_Device_DistributionImageFileID FOREIGN KEY (DistributionImageFileID) REFERENCES Dis-
tributionImageFile (DistributionImageFileID);
CONSTRAINT FK_Device_TimeZoneID FOREIGN KEY (TimeZoneID) REFERENCES TimeZone (TimeZoneID);
CONSTRAINT FK_Device_UpdateNotificationID FOREIGN KEY (UpdateNotificationID) REFERENCES UpdateNo-
tification (UpdateNotificationID);
CONSTRAINT FK_Device_DeliveryNotificationID FOREIGN KEY (DeliveryNotificationID) REFERENCES UpdateNo-
tification (UpdateNotificationID)
CONSTRAINT UQ_Device_ClientIdentifier UNIQUE (ClientIdentifier);
CONSTRAINT FK_Device_BiosUpdateNotificationID FOREIGN KEY (BiosUpdateNotificationID) REFERENCES UpdateNo-
tification (UpdateNotificationID);
PRIMARY KEY (DeviceID
```

3.4. Table: DeviceSchedulerJob

Configuration of the eLux Command Scheduler

ColumnName	DataType	Key	Not Null	Default Value	Comment
DeviceSchedulerJobID	UNIQUEIDENTIFIER	PK	NN		Unique ID
MemberType	INTEGER				OU ID or device ID
Command	INTEGER				Command ID **1** Update **2** SynchronizeConfiguration **3** Restart **4** User-defined **5** Shutdown
SpreadFactor	INTEGER				Selected spread factor
Persistent	BIT				Specifies whether the command continues to be executed after device restart
Wakeup	BIT				Specifies whether the system will be woken up from suspend mode to execute scheduled commands
DefermentCount	INTEGER				
DefermentTimes	NVARCHAR(255)				Depending on the (number of) deferments defined, the string contains the deferments in minutes such as "15,30,60,120,240,480" or "15,480"
UserCanCancel	BIT				**0|1**
UserConfirmation	BIT				**0|1**

ColumnName	DataType	Key	Not Null	Default Value	Comment
UserConfirmationTimeout	INTEGER				
RepeatingType	TINYINT				0 Daily 1 Weekly 2 Monthly 3 User-defined
OnlyWorkingDays	BIT				0|1
[WeekDay]	TINYINT				0 Monday 1 Tuesday 2 Wednesday 3 Thursday 4 Friday 5 Saturday 6 Sunday
MonthDay	TINYINT				1 First day of month 2 Second day of month … … 32 Last day of month
UserDefinedCommand	NVARCHAR(4000)				Only for user-defined commands: Command string
TimeScheduled	INTEGER				
ScheduleString	NVARCHAR(255)				String for user-defined repeating type

IndexName	IndexType	Columns
PRIMARY	PRIMARY	DeviceSchedulerJobID

unicon

How to create

```
CREATE TABLE DeviceSchedulerJob(
DeviceSchedulerJobID UNIQUEIDENTIFIER NOT NULL,
MemberType INTEGER,
MemberID INTEGER,
Command INTEGER,
SpreadFactor INTEGER,
Persistent BIT,
Wakeup BIT,
DefermentCount INTEGER,
DefermentTimes NVARCHAR(255),
UserCanCancel BIT,
UserConfirmation BIT,
UserConfirmationTimeout INTEGER,
RepeatingType TINYINT,
OnlyWorkingDays BIT,
[WeekDay] TINYINT,
MonthDay TINYINT
UserDefinedCommand NVARCHAR(4000)
TimeScheduled INTEGER,
ScheduleString NVARCHAR(255)
CONSTRAINT PK_DeviceSchedulerJob
PRIMARY KEY (DeviceSchedulerJobID))
```

3.5. Table: DistributionImageFile

This table holds all image file names which are reported from the clients. If one of the image names are marked as outdated, the Scout Enterprise Dashboard is not able to assign this image file any longer.

ColumnName	DataType	Key	Not Null	Default Value	Comment
DistributionImageFileID	INTEGER	PK	NN		Unique ID
ImageFile	NVARCHAR (255)		NN		Name of image file
Outdated	BIT		NN		Outdated status
Modified	DATETIME		NN		Last modification of this entry

IndexName	IndexType	Columns
PRIMARY	PRIMARY	DistributionImageFileID

How to create

```
CREATE TABLE DistributionImageFile(
 DistributionImageFileID INTEGER NOT NULL,
 ImageFile NVARCHAR(255),
 Outdated BIT NOT NULL,
 Modified DATETIME NOT NULL,
 CONSTRAINT PK_DistributionImageFile,
 PRIMARY KEY (DistributionImageFile));
```

3.6. Table: EnvironmentEntry

The entry represents a environment variable which is set on the device. It can be assigned to a single device or group.

ColumnName	DataType	Key	Not Null	Default Value	Comment
EnvironmentEntryID	INTEGER	PK	NN		Unique environment entry ID
MemberType	INTEGER				Member type **1** TYPE_GROUP MemberID refers to Groups **4** TYPE_DEVICE_ENTRY MemberID refers to Device
MemberID	INTEGER	(FK)	NN		MemberID (depending on MemberType this refers to table Groups or Device)
EnvironmentKey	NVARCHAR (255)				Name of environment variable
EnvironmentValue	NVARCHAR (255)				Value of environment variable
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	EnvironmentEntryID MemberID

How to create

```
CREATE TABLE EnvironmentEntry(
 EnvironmentEntryID INTEGER NOT NULL,
 MemberType INTEGER,
 MemberID INTEGER,
 EnvironmentKey NVARCHAR(255),
 EnvironmentValue NVARCHAR(255),
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (EnvironmentEntryID));
```

3.7. Table: FileEntry

For file transfer purposes, an entry in this table specifies the file which is transferred from the Scout Enterprise server to the thin client. The file transfer can be assigned to a single device, an organization unit, or at enterprise level (Advanced options).

ColumnName	DataType	Key	Not Null	Default Value	Comment
FileEntryID	INTEGER	PK	NN		Unique file entry ID
MemberType	INTEGER				Member type **-1** Enterprise — MemberID must be -1 **1** TYPE_GROUP — MemberID refers to Groups **4** TYPE_DEVICE_ENTRY — MemberID refers to Device
MemberID	INTEGER	(FK)	NN		Member ID (depending on MemberType, this refers to table Groups or Device)
FileSource	NVARCHAR(255)				Source file to be transferred to the client
FileDestination	NVARCHAR(255)				Target file on the client
Description	NVARCHAR(255)				Contains the description if the file entry is a destination file template on the client
UCFTID	INTEGER				This field refers to an entry in the UCFT table (for files that have been imported to the database).
ReservedString1	NVARCHAR(255)				

ColumnName	DataType	Key	Not Null	Default Value	Comment
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	FileEntryID MemberID

How to create

```
CREATE TABLE FileEntry(
 FileEntryID INTEGER NOT NULL,
 MemberType INTEGER,
 MemberID INTEGER,
 FileSource NVARCHAR(255),
 FileDestination NVARCHAR(255),
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (FileEntryID));
```

3.8. Table: FirefoxDefaults

This table stores the Firefox software defaults.

ColumnName	DataType	Key	Not Null	Default Value	Comment
FirefoxDefaultsID	INTEGER IDENTITY (1,1)	PK	NN		Unique ID
HomeDirectory	NVARCHAR(255)				holds the home directory
ItemID	INTEGER		NN		OU ID
ItemType	INTEGER		NN		always 1 (TYPE_OU) at this time

IndexName	IndexType	Columns
PRIMARY	PRIMARY	FirefoxDefaultsID

How to create

```
CREATE TABLE FirefoxDefaults(
FirefoxDefaultsID INTEGER IDENTITY (1,1) NOT NULL,
HomeDirectory NVARCHAR(255),
ItemID INTEGER NOT NULL,
ItemType INTEGER NOT NULL,
CONSTRAINT PK_FirefoxDefaults,
PRIMARY KEY (FirefoxDefaultsID));
```

How to initialize

```
INSERT INTO FirefoxDefaults(HomeDirectory, ItemType, ItemID) SELECT
Home,1,-1 FROM Setup WHERE SetupID=0;
```

3.9. Table: Groups

Each entry represents a organization unit (formerly known as group) of the Scout Enterprise hierarchy. Every OU has exact one owner OU, which is also known as parent OU.

ColumnName	DataType	Key	Not Null	Default Value	Comment
GroupID	INTEGER	PK	NN		Unique OU/group ID
WolServerID	INTEGER	FK			WakeOnLAN server ID Foreign key referring to table Server.ServerID
LocationID	INTEGER	FK	NN		ID of the owning OU/group Foreign key refers to table Groups.GroupID **-1**: top level OU
Name	NVARCHAR (255)		NN		Unique name of OU/group
UseParent	SMALLINT(6)		NN	1	Use setup of the owning OU/group **0|1**
Info1	NVARCHAR (255)				Arbitrary user information
Info2	NVARCHAR (255)				Arbitrary user information
Info3	NVARCHAR (255)				Arbitrary user information

ColumnName	DataType	Key	Not Null	Default Value	Comment
ClientAccessType	INTEGER			1	Group access control for clients **0** OU/group is invisible to clients (not browsable) **1** OU/group is visible (browsable and joinable)
ClientAccessPassword	NVARCHAR (255)				Crypted password required to join the OU/group
UseParentApps	SMALLINT(6)			1	Use applications of the owning group in addition **0|1**
UseParentAppDefaults	SMALLINT(6)			1	Use application settings of the owning OU/group **0|1**
DefaultPrinter	NVARCHAR (255)				Name of default printer (must exist in table Printer for the according SetupID)
DriverNames	NVARCHAR (max)				OU/group specific list of printer/driver mappings `printer1==driver1::printer2==driver2::printerN==driverN`
LastApplicationAction	NVARCHAR (255)			#NONE	Action takes place when the last application is closed #NONE — Do nothing #RESTARTX — Logoff user #REBOOT — Reboot device #OFF — Shutdown the device #LOCK — Lock the device #VPNDISCON — Disconnect from VPN

ColumnName	DataType	Key	Not Null	Default Value	Comment
LastApplicationActionDelay	INTEGER			0	Delay (in seconds) before LastApplicationAction is processed
Resolution	NVARCHAR (255)			1024x768	Screen resolution 640x480|800x600|1024x768|1152x864|1280x1024|1600x1200
Frequency	INTEGER			60	Screen frequency 60|70|75|80|85|90|100
ColorDepth	INTEGER			16	Screen color depth 8|16|24
UseDDC	SMALLINT(6)			1	Use DDC setting of the owning OU/group 0|1
UseParentScreenSettings	SMALLINT(6)			1	Use screen settings of the owning OU/group 0|1
UpdateParallel	INTEGER			10	Amount of devices which are updated together
UpdateDelay	INTEGER			30	-obsolete-
UpdateConnect	INTEGER			2	Timeout (in seconds) when connecting a device
UseParentUpdateSettings	SMALLINT(6)			1	Use update settings of the owning OU/group 0|1
UseParentMouseKBSettings	SMALLINT(6)			1	Use mouse and keyboard settings of the owning OU/group 0|1

ColumnName	DataType	Key	Not Null	Default Value	Comment
UsePeerIP	SMALLINT(6)			0	Use peer IP (such as access point with NAT) when connecting the client (instead of device IP) **0|1**
UseParentArray	NVARCHAR (255)			FF	Bit field indicates which property should be used from the owning OU/group
FirstContactAction	INTEGER			-1	
SetupID	INTEGER	FK		0	Setup ID (foreign key referring to table Setup.SetupID)
DevNameTemplate	NVARCHAR (255)				Name of the device name template Example: <DEV>_MAC
DeviceCount	INTEGER				Number of devices in this OU and subordinate OUs
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				
Note	NVARCHAR (255)				
OUType	INTEGER		NN	0	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	GroupID

unicon

How to create

```
CREATE TABLE Groups(
 GroupID INTEGER NOT NULL,
 WolServerID INTEGER,
 LocationID INTEGER NOT NULL,
 Name NVARCHAR(255) NOT NULL,
 UseParent SMALLINT NOT NULL,
 Info1 NVARCHAR(255),
 Info2 NVARCHAR(255),
 Info3 NVARCHAR(255),
 ClientAccessType INTEGER,
 ClientAccessPassword NVARCHAR(255),
 UseParentApps SMALLINT,
 UseParentAppDefaults SMALLINT,
 DefaultPrinter NVARCHAR(255),
 DriverNames NVARCHAR(max),
 LastApplicationAction NVARCHAR(255),
 LastApplicationActionDelay INTEGER,
 Resolution NVARCHAR(255),
 Frequency INTEGER,
 ColorDepth INTEGER,
 UseParentScreenSettings SMALLINT,
 UpdateParallel INTEGER,
 UpdateDelay INTEGER,
 UpdateConnect INTEGER,
 UseParentUpdateSettings SMALLINT,
 UseParentMouseKBSettings SMALLINT,
 UsePeerIP SMALLINT,
```

```
UseParentArray NVARCHAR(255),
FirstContactAction INTEGER,
SetupID INTEGER,
DevNameTemplate NVARCHAR(255),
DeviceCount INTEGER,
ReservedString1 NVARCHAR(255),
ReservedString2 NVARCHAR(255),
ReservedInt1 INTEGER,
ReservedInt2 INTEGER,
Note NVARCHAR(255),
OUType INTEGER NOT NULL,
CONSTRAINT FK_Groups_WolServerID_Server_ServerID FOREIGN KEY (WolServerID) REFERENCES Server (ServerID),
CONSTRAINT FK_Groups_SetupID_Setup_SetupID FOREIGN KEY (SetupID) REFERENCES Setup (SetupID),
PRIMARY KEY (GroupID));
```

3.10. Table: ICADefaults

The ICA default settings can be assigned to a group or to the base (global) settings.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ICADefaultsID	INTEGER	PK	NN		Unique ID of ICA default settings
ItemID	INTEGER	PK	NN		Item ID specifies where this entry belongs to (foreign key refers to table Groups)
ItemType	INTEGER		NN		**1** TYPE_GROUP ItemID refers to Groups **11** TYPE_BASEAPPLICATION ItemID must be set to 0
TransferDefaults	SMALLINT(6)			1	Specifies if the settings should be transferred to the clients **0|1**
ClientNameTemplate	NVARCHAR(255)				Template to be used to create the client name
DesiredHRES	NVARCHAR(32)				Window width
DesiredVRES	NVARCHAR(32)				Window heigth
ScreenPercent	NVARCHAR(32)				Window size specified as percentage of screen size. DesiredHRES or DesiredVRES must not be specified, UseFullScreen must be set to **No**
NoWindowManager	NVARCHAR(16)			True	set by Scout Enterprise server

ColumnName	DataType	Key	Not Null	Default Value	Comment
UseFullScreen	NVARCHAR(16)				Window size is set to screen size. DesiredHRES, DesiredVRES or ScreenPercent must not be specified **Yes|No**
DesiredColor	NVARCHAR(16)			4	Color depth **1** 4 bit (16 colors) **2** 8 bit (256 colors) **4** 16 bit (32 thousand colors) **8** 24 bit (16 million colors)
UseDefaultSettingFor Colormap	NVARCHAR(16)			No	set by Scout Enterprise server
ApproximateColors	NVARCHAR(16)			Yes	set by Scout Enterprise server
TW2StopwatchMinimum	INTEGER			25	Data flow control timer
PersistentCachePath	NVARCHAR(255)			/tmp	Directory where the bitmap cache resides
PersistentCacheSize	INTEGER			0	Bitmap cache size (max. 256*1024*1024=268435456)
PersistentCachePercent	INTEGER			0	Bitmap cache size in percent of available space
PersistentCacheMinBitmap	INTEGER			2048	Minimum bitmap size for caching
DisableSound	NVARCHAR(255)			True	Disable Windows alert sounds **True|False**

ColumnName	DataType	Key	Not Null	Default Value	Comment
ClientManagement	NVARCHAR(255)			Off	Allow automatic client updates **On|Off**
ClientDrive	NVARCHAR(255)			On	Drive mapping enabled **On|Off**
BrowserProtocol	NVARCHAR(255)				Protocol used to browse published applications **UDP|HTTPonTCP**
DoNotUseDefaultCSL	NVARCHAR(16)			On	obsolete
SSLEnable	NVARCHAR(16)				Use SSL and HTTPS (available for BrowserProtocol=HTTPonTCP only) **On|Off**
TcpGroupName1	NVARCHAR(255)				Name of group 1 (Primary)
TcpGroupName2	NVARCHAR(255)				Name of group 2 (Backup 1)
TcpGroupName3	NVARCHAR(255)				Name of group 3 (Backup 2)
TransportReconnectEnabled	NVARCHAR(255)			True	Enable reconnect **True|False**
TransportReconnectRetries	INTEGER			3	Reconnect retries
TransportReconnectDelay	INTEGER			30	Reconnect delay
UseAlternateAddress	SMALLINT(6)			0	Use alternate address for firewall connection **0|1**

ColumnName	DataType	Key	Not Null	Default Value	Comment
ProxyHost	NVARCHAR(255)				Connect via proxy server (ProxyType must be set) <proxy>:<port>
ProxyType	NVARCHAR(255)			None	Type of proxy server **None|Auto|Socks|Secure**
ServerURL	NVARCHAR(255)				set by Scout Enterprise server
HttpBrowserAddress	NVARCHAR(255)				ICA browser ip address of group 1 (if BrowserProtocol=**HTTPonTCP)**
HttpBrowserAddress2	NVARCHAR(255)				see HttpBrowserAddress
HttpBrowserAddress3	NVARCHAR(255)				see HttpBrowserAddress
HttpBrowserAddress4	NVARCHAR(255)				see HttpBrowserAddress
HttpBrowserAddress5	NVARCHAR(255)				see HttpBrowserAddress
HttpBrowserAddress6	NVARCHAR(255)				ICA browser ip address of group 2 (if BrowserProtocol=**HTTPonTCP)**
HttpBrowserAddress7	NVARCHAR(255)				see HttpBrowserAddress6
HttpBrowserAddress8	NVARCHAR(255)				see HttpBrowserAddress6
HttpBrowserAddress9	NVARCHAR(255)				see HttpBrowserAddress6
HttpBrowserAddress10	NVARCHAR(255)				see HttpBrowserAddress6
HttpBrowserAddress11	NVARCHAR(255)				ICA browser ip address of group 3 (if BrowserProtocol=**HTTPonTCP)**
HttpBrowserAddress12	NVARCHAR(255)				see HttpBrowserAddress11

unicon

ColumnName	DataType	Key	Not Null	Default Value	Comment
HttpBrowserAddress13	NVARCHAR(255)				see HttpBrowserAddress11
HttpBrowserAddress14	NVARCHAR(255)				see HttpBrowserAddress11
HttpBrowserAddress15	NVARCHAR(255)				see HttpBrowserAddress11
TcpBrowserAddress	NVARCHAR(255)				ICA browser ip address of group 1 (if BrowserProtocol=**UDP)**
TcpBrowserAddress2	NVARCHAR(255)				see TcpBrowserAddress
TcpBrowserAddress3	NVARCHAR(255)				see TcpBrowserAddress
TcpBrowserAddress4	NVARCHAR(255)				see TcpBrowserAddress
TcpBrowserAddress5	NVARCHAR(255)				see TcpBrowserAddress
TcpBrowserAddress6	NVARCHAR(255)				ICA browser ip address of group 2 (if BrowserProtocol=**UDP)**
TcpBrowserAddress7	NVARCHAR(255)				see TcpBrowserAddress6
TcpBrowserAddress8	NVARCHAR(255)				see TcpBrowserAddress6
TcpBrowserAddress9	NVARCHAR(255)				see TcpBrowserAddress6
TcpBrowserAddress10	NVARCHAR(255)				see TcpBrowserAddress6
TcpBrowserAddress11	NVARCHAR(255)				ICA browser ip address of group 3 (if BrowserProtocol=**UDP)**
TcpBrowserAddress12	NVARCHAR(255)				see TcpBrowserAddress11
TcpBrowserAddress13	NVARCHAR(255)				see TcpBrowserAddress11

ColumnName	DataType	Key	Not Null	Default Value	Comment
TcpBrowserAddress14	NVARCHAR(255)				see TcpBrowserAddress11
TcpBrowserAddress15	NVARCHAR(255)				see TcpBrowserAddress11
Hotkey1Char	NVARCHAR(16)			(None)	Hotkey name (used with modifier Hotkey1Shift) (None)|F1|F2|F3|F4|F5|F6|F7|F8|F9|F10|F11|F12|Tab|Minus|Plus
Hotkey1Shift	NVARCHAR(16)			(None)	Modifier key used in conjunction with Hotkey1Char (None)|Shift|Ctrl|Alt|Alt-Ctrl|Alt-Shift|Ctrl-Alt
Hotkey2Char	NVARCHAR(16)				see Hotkey1Char
Hotkey2Shift	NVARCHAR(16)				see Hotkey1Shift
Hotkey3Char	NVARCHAR(16)				see Hotkey1Char
Hotkey3Shift	NVARCHAR(16)				see Hotkey1Shift
Hotkey4Char	NVARCHAR(16)				see Hotkey1Char
Hotkey4Shift	NVARCHAR(16)				see Hotkey1Shift
Hotkey5Char	NVARCHAR(16)				see Hotkey1Char
Hotkey5Shift	NVARCHAR(16)				see Hotkey1Shift
Hotkey6Char	NVARCHAR(16)				see Hotkey1Char
Hotkey6Shift	NVARCHAR(16)				see Hotkey1Shift
Hotkey7Char	NVARCHAR(16)				see Hotkey1Char

ColumnName	DataType	Key	Not Null	Default Value	Comment
Hotkey7Shift	NVARCHAR(16)				see Hotkey1Shift
Hotkey8Char	NVARCHAR(16)				see Hotkey1Char
Hotkey8Shift	NVARCHAR(16)				see Hotkey1Shift
Hotkey9Char	NVARCHAR(16)				see Hotkey1Char
Hotkey9Shift	NVARCHAR(16)				see Hotkey1Shift
Hotkey10Char	NVARCHAR(16)				see Hotkey1Char
Hotkey10Shift	NVARCHAR(16)				see Hotkey1Shift
Hotkey11Char	NVARCHAR(16)				see Hotkey1Char
Hotkey11Shift	NVARCHAR(16)				see Hotkey1Shift
Hotkey12Char	NVARCHAR(16)				see Hotkey1Char
Hotkey12Shift	NVARCHAR(16)				see Hotkey1Shift
Hotkey13Char	NVARCHAR(16)				see Hotkey1Char
Hotkey13Shift	NVARCHAR(16)				see Hotkey1Shift
Hotkey14Char	NVARCHAR(16)				see Hotkey1Char
Hotkey14Shift	NVARCHAR(16)				see Hotkey1Shift
Hotkey15Char	NVARCHAR(16)				see Hotkey1Char

ColumnName	DataType	Key	Not Null	Default Value	Comment
Hotkey15Shift	NVARCHAR(16)				see Hotkey1Shift
DrivePathA	NVARCHAR(max)				Client directory mapped to a drive in server session, holds definition for all drive letters e.g ###/media/usb0###/media/usb1######/media/usb2 where A is /media/usb0, B is /media/usb1 and D is /media/usb2
DriveEnabledA	NVARCHAR(255)			Yes	Mapping enabled for drive all drive letters **e.g ###Yes###Yes###No###Yes**
DriveReadAccessA	SMALLINT(6)			0	Read access restriction **0** Read access is unrestricted **1** Read access denied **2** Ask user if read access should be granted
DriveWriteAccessA	SMALLINT(6)			0	Write access restriction **0** Write access is unrestricted **1** Write access denied **2** Ask user if write access should be granted
DriveReadAccessB	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessB	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessC	SMALLINT(6)			0	see DriveReadAccessA

unicon

ColumnName	DataType	Key	Not Null	Default Value	Comment
DriveWriteAccessC	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessD	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessD	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessE	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessE	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessF	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessF	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessG	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessG	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessH	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessH	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessI	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessI	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessJ	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessJ	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessK	SMALLINT(6)			0	see DriveReadAccessA

unicon

ColumnName	DataType	Key	Not Null	Default Value	Comment
DriveWriteAccessK	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessL	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessL	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessM	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessM	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessN	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessN	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessO	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessO	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessP	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessP	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessQ	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessQ	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessR	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessR	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessS	SMALLINT(6)			0	see DriveReadAccessA

ColumnName	DataType	Key	Not Null	Default Value	Comment
DriveWriteAccessS	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessT	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessT	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessU	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessU	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessV	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessV	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessW	SMALLINT(6)			0	see DriveReadAccessA
DriveWriteAccessW	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessX	SMALLINT(6)			No	see DriveReadAccessA
DriveWriteAccessX	SMALLINT(6)			0	see DriveWriteAccessA
DriveReadAccessY	SMALLINT(6)			No	see DriveReadAccessA
DriveWriteAccessY	SMALLINT(6)				see DriveWriteAccessA
DriveReadAccessZ	SMALLINT(6)			No	see DriveReadAccessA
DriveWriteAccessZ	SMALLINT(6)				see DriveWriteAccessA
COM1	NVARCHAR(255)			/dev/ttyS0	Driver interface mapped to COM1

ColumnName	DataType	Key	Not Null	Default Value	Comment
COM2	NVARCHAR(255)				Driver interface mapped to COM2
COM3	NVARCHAR(255)				Driver interface mapped to COM3
COM4	NVARCHAR(255)				Driver interface mapped to COM4
COM5	NVARCHAR(255)				Driver interface mapped to COM5
AllowAudioInput	NVARCHAR(16)				
TransparentKeyPassthrough	NVARCHAR(255)				
KeyPassthroughEscapeChar	NVARCHAR(16)				
KeyPassthroughEscapeShift	NVARCHAR(16)				
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ICADefaultsID ItemID

unicon

How to create

```
CREATE TABLE ICADefaults(
 ICADefaultsID INTEGER NOT NULL,
 ItemID INTEGER NOT NULL,
 ItemType INTEGER NOT NULL,
 TransferDefaults SMALLINT,
 ClientNameTemplate NVARCHAR(255),
 DesiredHRES NVARCHAR(32),
 DesiredVRES NVARCHAR(32),
 ScreenPercent NVARCHAR(32),
 NoWindowManager NVARCHAR(16),
 UseFullScreen NVARCHAR(16),
 DesiredColor NVARCHAR(16),
 UseDefaultSettingForColormap NVARCHAR(16),
 ApproximateColors NVARCHAR(16),
 TW2StopwatchMinimum INTEGER,
 PersistentCachePath NVARCHAR(255),
 PersistentCacheSize INTEGER,
 PersistentCachePercent INTEGER,
 PersistentCacheMinBitmap INTEGER,
 DisableSound NVARCHAR(255),
 ClientManagement NVARCHAR(255),
 ClientDrive NVARCHAR(255),
 BrowserProtocol NVARCHAR(255),
 DoNotUseDefaultCSL NVARCHAR(16),
 SSLEnable NVARCHAR(16),
 TcpGroupName1 NVARCHAR(255),
 TcpGroupName2 NVARCHAR(255),
```

```
TcpGroupName3 NVARCHAR(255),
TransportReconnectEnabled NVARCHAR(255),
TransportReconnectRetries INTEGER,
TransportReconnectDelay INTEGER,
UseAlternateAddress SMALLINT,
ProxyHost NVARCHAR(255),
ProxyType NVARCHAR(255),
ServerURL NVARCHAR(255),
HttpBrowserAddress NVARCHAR(255),
HttpBrowserAddress2 NVARCHAR(255),
HttpBrowserAddress3 NVARCHAR(255),
HttpBrowserAddress4 NVARCHAR(255),
HttpBrowserAddress5 NVARCHAR(255),
HttpBrowserAddress6 NVARCHAR(255),
HttpBrowserAddress7 NVARCHAR(255),
HttpBrowserAddress8 NVARCHAR(255),
HttpBrowserAddress9 NVARCHAR(255),
HttpBrowserAddress10 NVARCHAR(255),
HttpBrowserAddress11 NVARCHAR(255),
HttpBrowserAddress12 NVARCHAR(255),
HttpBrowserAddress13 NVARCHAR(255),
HttpBrowserAddress14 NVARCHAR(255),
HttpBrowserAddress15 NVARCHAR(255),
TcpBrowserAddress NVARCHAR(255),
TcpBrowserAddress2 NVARCHAR(255),
TcpBrowserAddress3 NVARCHAR(255),
TcpBrowserAddress4 NVARCHAR(255),
TcpBrowserAddress5 NVARCHAR(255),
TcpBrowserAddress6 NVARCHAR(255),
```

```
TcpBrowserAddress7 NVARCHAR(255),
TcpBrowserAddress8 NVARCHAR(255),
TcpBrowserAddress9 NVARCHAR(255),
TcpBrowserAddress10 NVARCHAR(255),
TcpBrowserAddress11 NVARCHAR(255),
TcpBrowserAddress12 NVARCHAR(255),
TcpBrowserAddress13 NVARCHAR(255),
TcpBrowserAddress14 NVARCHAR(255),
TcpBrowserAddress15 NVARCHAR(255),
Hotkey1Char NVARCHAR(16),
Hotkey1Shift NVARCHAR(16),
Hotkey2Char NVARCHAR(16),
Hotkey2Shift NVARCHAR(16),
Hotkey3Char NVARCHAR(16),
Hotkey3Shift NVARCHAR(16),
Hotkey4Char NVARCHAR(16),
Hotkey4Shift NVARCHAR(16),
Hotkey5Char NVARCHAR(16),
Hotkey5Shift NVARCHAR(16),
Hotkey6Char NVARCHAR(16),
Hotkey6Shift NVARCHAR(16),
Hotkey7Char NVARCHAR(16),
Hotkey7Shift NVARCHAR(16),
Hotkey8Char NVARCHAR(16),
Hotkey8Shift NVARCHAR(16),
Hotkey9Char NVARCHAR(16),
Hotkey9Shift NVARCHAR(16),
Hotkey10Char NVARCHAR(16),
Hotkey10Shift NVARCHAR(16),
```

```
Hotkey11Char NVARCHAR(16),
Hotkey11Shift NVARCHAR(16),
Hotkey12Char NVARCHAR(16),
Hotkey12Shift NVARCHAR(16),
Hotkey13Char NVARCHAR(16),
Hotkey13Shift NVARCHAR(16),
Hotkey14Char NVARCHAR(16),
Hotkey14Shift NVARCHAR(16),
Hotkey15Char NVARCHAR(16),
Hotkey15Shift NVARCHAR(16),
DrivePathA NVARCHAR(255),
DriveEnabledA NVARCHAR(255),
DriveReadAccessA SMALLINT,
DriveWriteAccessA SMALLINT,
DrivePathB NVARCHAR(255),
DriveEnabledB NVARCHAR(255),
DriveReadAccessB SMALLINT,
DriveWriteAccessB SMALLINT,
DrivePathC NVARCHAR(255),
DriveEnabledC NVARCHAR(255),
DriveReadAccessC SMALLINT,
DriveWriteAccessC SMALLINT,
DrivePathD NVARCHAR(255),
DriveEnabledD NVARCHAR(255),
DriveReadAccessD SMALLINT,
DriveWriteAccessD SMALLINT,
DrivePathE NVARCHAR(255),
DriveEnabledE NVARCHAR(255),
DriveReadAccessE SMALLINT,
```

```
DriveWriteAccessE SMALLINT,
DrivePathF NVARCHAR(255),
DriveEnabledF NVARCHAR(255),
DriveReadAccessF SMALLINT,
DriveWriteAccessF SMALLINT,
DrivePathG NVARCHAR(255),
DriveEnabledG NVARCHAR(255),
DriveReadAccessG SMALLINT,
DriveWriteAccessG SMALLINT,
DrivePathH NVARCHAR(255),
DriveEnabledH NVARCHAR(255),
DriveReadAccessH SMALLINT,
DriveWriteAccessH SMALLINT,
DrivePathI NVARCHAR(255),
DriveEnabledI NVARCHAR(255),
DriveReadAccessI SMALLINT,
DriveWriteAccessI SMALLINT,
DrivePathJ NVARCHAR(255),
DriveEnabledJ NVARCHAR(255),
DriveReadAccessJ SMALLINT,
DriveWriteAccessJ SMALLINT,
DrivePathK NVARCHAR(255),
DriveEnabledK NVARCHAR(255),
DriveReadAccessK SMALLINT,
DriveWriteAccessK SMALLINT,
DrivePathL NVARCHAR(255),
DriveEnabledL NVARCHAR(255),
DriveReadAccessL SMALLINT,
DriveWriteAccessL SMALLINT,
```

```
DrivePathM NVARCHAR(255),
DriveEnabledM NVARCHAR(255),
DriveReadAccessM SMALLINT,
DriveWriteAccessM SMALLINT,
DrivePathN NVARCHAR(255),
DriveEnabledN NVARCHAR(255),
DriveReadAccessN SMALLINT,
DriveWriteAccessN SMALLINT,
DrivePathO NVARCHAR(255),
DriveEnabledO NVARCHAR(255),
DriveReadAccessO SMALLINT,
DriveWriteAccessO SMALLINT,
DrivePathP NVARCHAR(255),
DriveEnabledP NVARCHAR(255),
DriveReadAccessP SMALLINT,
DriveWriteAccessP SMALLINT,
DrivePathQ NVARCHAR(255),
DriveEnabledQ NVARCHAR(255),
DriveReadAccessQ SMALLINT,
DriveWriteAccessQ SMALLINT,
DrivePathR NVARCHAR(255),
DriveEnabledR NVARCHAR(255),
DriveReadAccessR SMALLINT,
DriveWriteAccessR SMALLINT,
DrivePathS NVARCHAR(255),
DriveEnabledS NVARCHAR(255),
DriveReadAccessS SMALLINT,
DriveWriteAccessS SMALLINT,
DrivePathT NVARCHAR(255),
```

```
DriveEnabledT NVARCHAR(255),
DriveReadAccessT SMALLINT,
DriveWriteAccessT SMALLINT,
DrivePathU NVARCHAR(255),
DriveEnabledU NVARCHAR(255),
DriveReadAccessU SMALLINT,
DriveWriteAccessU SMALLINT,
DrivePathV NVARCHAR(255),
DriveEnabledV NVARCHAR(255),
DriveReadAccessV SMALLINT,
DriveWriteAccessV SMALLINT,
DrivePathW NVARCHAR(255),
DriveEnabledW NVARCHAR(255),
DriveReadAccessW SMALLINT,
DriveWriteAccessW SMALLINT,
DrivePathX NVARCHAR(255),
DriveEnabledX NVARCHAR(255),
DriveReadAccessX SMALLINT,
DriveWriteAccessX SMALLINT,
DrivePathY NVARCHAR(255),
DriveEnabledY NVARCHAR(255),
DriveReadAccessY SMALLINT,
DriveWriteAccessY SMALLINT,
DrivePathZ NVARCHAR(255),
DriveEnabledZ NVARCHAR(255),
DriveReadAccessZ SMALLINT,
DriveWriteAccessZ SMALLINT,
COM1 NVARCHAR(255),
COM2 NVARCHAR(255),
```

```
COM3 NVARCHAR(255),
COM4 NVARCHAR(255),
COM5 NVARCHAR(255),
AllowAudioInput NVARCHAR(16),
TransparentKeyPassthrough NVARCHAR(255),
KeyPassthroughEscapeChar NVARCHAR(16),
KeyPassthroughEscapeShift NVARCHAR(16),
ReservedString1 NVARCHAR(255),
ReservedString2 NVARCHAR(255),
ReservedInt1 INTEGER,
ReservedInt2 INTEGER,
PRIMARY KEY (ICADefaultsID));
```

3.11. Table: IniEntry

Each record represents a entry in a ini file which is located at the thin client. It can be assigned to a single device or group.

ColumnName	DataType	Key	Not Null	Default Value	Comment
IniEntryID	INTEGER	PK	NN		Unique ini entry ID
MemberType	INTEGER				Member type 1 TYPE_GROUP MemberID refers to Groups 4 TYPE_DEVICE_ENTRY MemberID refers to Device
MemberID	INTEGER	(FK)	NN		MemberID (depending on MemberType this refers to table Groups or Device)
IniFile	NVARCHAR (255)				Name of ini file (absolute path on thin client)
IniSection	NVARCHAR (255)				Name of section in ini file
IniKey	NVARCHAR (255)				Name of key in ini file
IniValue	NVARCHAR (255)				Value
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	IniEntryID MemberID

How to create

```
CREATE TABLE IniEntry(
 IniEntryID INTEGER NOT NULL,
 MemberType INTEGER,
 MemberID INTEGER,
 IniFile NVARCHAR(255),
 IniSection NVARCHAR(255),
 IniKey NVARCHAR(255),
 IniValue NVARCHAR(255),
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (IniEntryID));
```

3.12. Table: MouseKB

Each record represents a set of mouse and keyboard settings which are relevant to a single device or group.

ColumnName	DataType	Key	Not Null	DefaultValue	Comment
MouseKBID	INTEGER	PK	NN		Unique ini entry ID
MemberType	INTEGER				**1** TYPE_GROUP — MemberID refers to Groups **4** TYPE_DEVICE_ENTRY — MemberID refers to Device
MemberID	INTEGER	(FK)	NN		MemberID (depending on MemberType. this refers to table Groups or Device)
PointerProtocol	NVARCHAR(10)		NN	auto	Mouse pointer protocol **auto** auto-detect (Buttons=**5**) **PS/2** 2 button mouse (Buttons=**2**) or 3 button mouse (Buttons=**3**) **IMPS/2** wheel mouse (Buttons=**5**) **none** no mouse (set Buttons=**none**)
LeftHandMouse	SMALLINT(6)		NN	0	Left hand mouse **0|1**
MultiClickTime	INTEGER		NN	500	Double click time 100(fast) - 900(slow)
DeadKeys	SMALLINT(6)		NN	1	Dead keys active **0|1**
Numlock	SMALLINT(6)		NN	1	Numlock at start **0|1**

ColumnName	DataType	Key	Not Null	DefaultValue	Comment
PointerSpeed	NVARCHAR(15)			2/1	Mouse acceleration 3/10(slow) - 13/2(fast)
KeyboardDelay	INTEGER			500	Keyboard delay (100-1510)
KeyboardSpeed	INTEGER		NN	30	Keyboard speed (10-250)
ConsoleSwitch	INTEGER				
XkbLayoutID	INTEGER	FK	NN		Keyboard Layout (foreign key refers to table Xkblayout.XkbLayoutID)
KeyboardModel	NVARCHAR(255)				**Auto** Keyboard model is detected automatically: configured by default **Cherry** Cherry G81-8000 keyboard **Desko** Desko MCx 4717 keyboard **K293** MFII keyboard **TYPE6** Sun Type6 keyboard **K235** Trimo 3270 keyboard **K257** 97801 keyboard
ExtendedKeys	INTEGER				0: No keys 127: All keys
TouchpadState	TINYINT			1	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	MouseKBID MemberID

unicon

How to create

```
CREATE TABLE MouseKB(
 MouseKBID INTEGER NOT NULL,
 MemberType INTEGER NOT NULL,
 MemberID INTEGER NOT NULL,
 PointerProtocol NVARCHAR(10) NOT NULL,
 LeftHandMouse SMALLINT NOT NULL,
 MultiClickTime INTEGER NOT NULL,
 DeadKeys SMALLINT NOT NULL,
 Numlock SMALLINT NOT NULL,
 PointerSpeed NVARCHAR(15),
 KeyboardDelay INTEGER,
 KeyboardSpeed INTEGER NOT NULL,
 ConsoleSwitch INTEGER,
 XkbLayoutID INTEGER NOT NULL,
 KeyboardModel NVARCHAR(255),
 TouchpadState TINYINT,
 ExtendedKeys INTEGER,
 PRIMARY KEY (MouseKBID));
```

3.13. Table: PredefinedImagePath

Holds the data for predefined image paths in the firmware configuration.

ColumnName	DataType	Key	Not Null	Default Value	Comment
PredefinedImagePathID	INTEGER IDENTITY(1,1)	PK	NN		Unique ID of the path
ImagePath	NVARCHAR (255)		NN		Image path string
Outdated	BIT		NN		
Modified	DATETIME		NN		
Options	INTEGER			0	Predefined path type **0** Path for .idf **1** Path for .udf

IndexName	IndexType	Columns
PRIMARY	PRIMARY	PredefinedImagePathID

HOW TO CREATE

```
CREATE TABLE PredefinedImagePath(
PredefinedImagePathID INTEGER IDENTITY(1,1) NOT NULL,
ImagePath NVARCHAR(255) NOT NULL,
Outdated BIT NOT NULL,
Modified DATETIME NOT NULL,
Options INTEGER DEFAULT 0,
CONSTRAINT PK_PredefinedImagePath
PRIMARY KEY (PredefinedImagePathID));
```

3.14. Table: Relocation

If a device entry has an reference to an entry of this table, a relocation process will be initiated the next time the client boots.

ColumnName	DataType	Key	Not Null	Default Value	Comment
RelocationID	INTEGER	PK	NN		Unique ID
ServerID	INTEGER		NN		Reference to the destination server
Options	INTEGER		NN		Options for relocation
Modified	DATETIME		NN		Last modification of this entry
DestinationOU	NVARCHAR (255)				The destination OU to which the client should be added on the new Scout server

IndexName	IndexType	Columns
PRIMARY	PRIMARY	RelocationID

How to create

```
CREATE TABLE Relocation(
RelocationID INTEGER IDENTITY (1,1) NOT NULL,
ServerID INTEGER NOT NULL,
Options INTEGER NOT NULL,
Modified DATETIME NOT NULL,
DestinationOU NVARCHAR(255),
ReservedString1 NVARCHAR(255),
ReservedString2 NVARCHAR(255),
ReservedInt1 INTEGER,
ReservedInt2 INTEGER,
CONSTRAINT PK_Relocation
PRIMARY KEY (RelocationID));
```

3.15. Table: RequiredDeviceImageFile

This table defines whether a special image should be updated on the client. Normally, an image file is defined in the setup of a device or OU. To overwrite this configuration, an image file for a device can be defined here.

ColumnName	DataType	Key	Not Null	Default Value	Comment
RequiredDeviceImageFileID	INTEGER	PK	NN		Unique ID
DeviceID	INTEGER	FK	NN		ID of the device
DistributedImageFileID	INTEGER	FK	NN		Image file to be used on next update request

IndexName	IndexType	Column	References
PK_ RequiredDeviceImageFile	PRIMARY	DistributionImageFileID	
FK_ RequiredDeviceImageFile_ Device	FOREIGN	DeviceID	Device(DeviceID)
FK_ RequiredDeviceImageFile_ DistributedImageFile	FOREIGN	DistributedImageFileID	DistributionImageFile(Dis-tributionImageFileID)

How to create

```
CREATE TABLE RequiredDeviceImageFile(
 RequiredDeviceImageFileID INTEGER NOT NULL,
 DeviceID INTEGER,
 DistributedImageFileID INTEGER NOT NULL,
 CONSTRAINT FK_RequiredDeviceImageFile_Device FOREIGN KEY
(DeviceID) REFERENCES Device (DeviceID),
 CONSTRAINT FK_RequiredDeviceImageFile_RequiredDeviceImageFile
FOREIGN KEY (DistributedImageFileID) REFERENCES Dis-
tributionImageFile (DistributionImageFileID),
 CONSTRAINT PK_RequiredDeviceImageFile
 PRIMARY KEY (RequiredDeviceImageFile));
```

3.16. Table: Simple Device Identifier

Temporary device identifier for support cases

ColumnName	DataType	Key	Not Null	Default Value	Comment
ClientIdentifier	UNIQUEIDENTIFIER		NN		Client device Identifier
SimpleIdentifier	NVARCHAR(128)				Simple device identifier
ValidTill	INTEGER				Time stamp for validity of the simple device identifier

How to create

```
CREATE TABLE SimpleDeviceIdentifier(
ClientIdentifier UNIQUEIDENTIFIER NOT NULL,
SimpleIdentifier NVARCHAR(128),
ValidTill INTEGER
```

3.17. Table: Tenant

Service Provider mode only: Tenants may be assigned to devices.

ColumnName	DataType	Key	Not Null	Default Value	Comment
TenantID	UNIQUEIDENTIFIER	PK	NN		Unique ID
Name	NVARCHAR(255)				Name of tenant
Comment	NVARCHAR(max)				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	TenantID

How to create

```
CREATE TABLE Tenant(
TenantID UNIQUEIDENTIFIER NOT NULL,
NameNVARCHAR(255),
Comment NVARCHAR(max),
CONSTRAINT PK_Tenant
PRIMARY KEY (TenantID))
```

4. Application

4.1. Table: Application

Each entry defines an application with its main attributes. All application specific data were derived to underlying tables.

ColumnName	DataType	Key	Not Null	Default	Comment
ApplicationID	INTEGER	PK	NN		Unique application ID
AppTypeID	INTEGER		NN		Application type 0 APPLTYPE_EMULATION Emulation 3 APPLTYPE_BROWSER Browser 4 APPLTYPE_ICA ICA client 5 APPLTYPE_LOCALSHELL Local application 7 APPLTYPE_SAPGUI SAP GUI 10 APPLTYPE_RDP RDP client 13 APPLTYPE_PNL PN agent 15 APPLTYPE_VD Virtual Desktop
Name	NVARCHAR(255)		NN		Application name The name must be unique for each combination of AppLevel and GroupID
AutoStart	SMALLINT(6)		NN		Start automatically 0|1
TM	SMALLINT(6)		NN		Application restart (terminal mode) 0|1

ColumnName	DataType	Key	Not Null	Default	Comment
AppOnDesktop	SMALLINT(6)				Desktop icon 0|1
Roaming	SMALLINT(6)		NN		Roaming (supported by AppTypeID=4 and AppTypeID=10) 0|1
AppLevel	SMALLINT(6)		NN		Hierarchical level of application 6 TYPE_APPLICATION_ENTRY OU/Group level
GroupID	INTEGER	(FK)	NN		ID of the OU/group which is owner of this application (see table Groups). For applications on base level this value must be set to -1.
IconName	NVARCHAR(255)				Name of the icon which to be displayed on the client).
PassThrough	SMALLINT(6)		NN	0	Defines whether an application can use pass-through login (supported by AppTypeID=4 and AppTypeID=13) 0|1
Xinerama	INTEGER		NN	0	Specifies the xinerama mode to be activated -1|0|1|2|3|4
SortID	INTEGER		NN		
LooseParameters	NVARCHAR(max)				Holds the free parameters of an application. Multiple parameters are separated by an '\r' character. Example: UseICAPreLogin=1
DisplayName	NVARCHAR(255)				Display name on client

ColumnName	DataType	Key	Not Null	Default	Comment
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				Start delay in seconds (used only if AutoStart is activated)
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationID

How to create

```
CREATE TABLE Application(
 ApplicationID INTEGER NOT NULL;
 AppTypeID INTEGER NOT NULL;
 Name NVARCHAR(255) NOT NULL;
 AutoStart SMALLINT NOT NULL;
 TM SMALLINT NOT NULL;
 AppOnDesktop SMALLINT;
 Roaming SMALLINT NOT NULL;
 AppLevel SMALLINT NOT NULL;
 GroupID INTEGER NOT NULL;
 IconName NVARCHAR(255);
 PassThrough SMALLINT;
 Xinerama INTEGER;
 SortID INTEGER NOT NULL CONSTRAINT DF_Application_SortID DEFAULT 1;
 LooseParameters NVARCHAR(max);
 DisplayName NVARCHAR(255);
 ReservedString1 NVARCHAR(255);
 ReservedString2 NVARCHAR(255);
 ReservedInt1 INTEGER;
 ReservedInt2 INTEGER;
 PRIMARY KEY (ApplicationID));
```

4.2. Table: ApplicationTemplate

To allow working with application templates, application template tables are provided. They contain the same fields as the application tables.

ColumnName	DataType	Key	Not Null	Default	Comment
ApplicationTemplateID	INTEGER	PK	NN		Unique application template ID
AppTypeID	INTEGER		NN		Application type 0 APPLTYPE_EMULATION Emulation 3 APPLTYPE_BROWSER Browser 4 APPLTYPE_ICA ICA client 5 APPLTYPE_LOCALSHELL Local application 7 APPLTYPE_SAPGUI SAP GUI 10 APPLTYPE_RDP RDP client 13 APPLTYPE_PNL PN agent 15 APPLTYPE_VD Virtual Desktop
Name	NVARCHAR(255)		NN		Application name The name must be unique for each combination of AppLevel and GroupID
AutoStart	SMALLINT(6)		NN		Start automatically 0|1
TM	SMALLINT(6)		NN		Application restart (terminal mode) 0|1
AppOnDesktop	SMALLINT(6)				Desktop icon 0|1
Roaming	SMALLINT(6)		NN		Roaming (supported by AppTypeID=4 and AppTypeID=10) 0|1

ColumnName	DataType	Key	Not Null	Default	Comment
AppLevel	SMALLINT(6)		NN		Hierarchical level of application 6 TYPE_APPLICATION_ENTRY OU/Group level
GroupID	INTEGER	(FK)	NN		ID of the OU/group which is owner of this application (see table Groups). For applications on base level this value must be set to -1.
IconName	NVARCHAR(255)				Name of the icon which to be displayed on the client).
PassThrough	SMALLINT(6)		NN	0	Defines whether an application can use pass-through login (supported by AppTypeID=4 and AppTypeID=13) **0|1**
Xinerama	INTEGER		NN	0	Specifies the xinerama mode to be activated **-1|0|1|2|3|4**
SortID	INTEGER		NN		
LooseParameters	NVARCHAR(max)				Holds the free parameters of an application. Multiple parameters are separated by an '\r' character. Example: UseICAPreLogin=1
DisplayName	NVARCHAR(255)				Display name on client
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				Start delay in seconds (used only if AutoStart is activated)
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationTemplateID

unicon

How to create

```
CREATE TABLE ApplicationTemplate(
 ApplicationTemplateID INTEGER NOT NULL;
 AppTypeID INTEGER NOT NULL;
 Name NVARCHAR(255) NOT NULL;
 AutoStart SMALLINT NOT NULL;
 TM SMALLINT NOT NULL;
 AppOnDesktop SMALLINT;
 Roaming SMALLINT NOT NULL;
 AppLevel SMALLINT NOT NULL;
 GroupID INTEGER NOT NULL;
 IconName NVARCHAR(255);
 PassThrough SMALLINT;
 Xinerama INTEGER;
 SortID INTEGER NOT NULL CONSTRAINT DF_Application_SortID DEFAULT 1;
 LooseParameters NVARCHAR(max);
 DisplayName NVARCHAR(255);
 ReservedString1 NVARCHAR(255);
 ReservedString2 NVARCHAR(255);
 ReservedInt1 INTEGER;
 ReservedInt2 INTEGER;
 CONSTRAINT PK_ApplicationTemplate;
 PRIMARY KEY (ApplicationTemplateID));
```

4.3. Table: Browser

Each entry describes an application of type APPLTYPE_BROWSER.

ColumnName	DataType	Key	Not Null	DefaultValue	Comment
ApplicationID	INTEGER	PK, FK	NN		Unique application ID Foreign key refers to table Application
StartPage	NVARCHAR (255)				Home page
CallPage	NVARCHAR (255)				Called at browser start
Proxy	NVARCHAR (255)				Proxy server (IP address or name)
ProxyType	INTEGER				Use proxy server **0|1**
BrowserType	INTEGER				**3** Firefox **4** Chromium
KioskMode	SMALLINT(6)				Kiosk mode **0|1**
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				
ProxyExceptions	NVARCHAR (max)				Holds the entries of the browser proxy exception list

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationID

How to create

```
CREATE TABLE Browser(
 ApplicationID INTEGER NOT NULL,
 StartPage NVARCHAR(255),
 CallPage NVARCHAR(255),
 Proxy NVARCHAR(255),
 ProxyType INTEGER,
 BrowserType INTEGER,
 KioskMode SMALLINT,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 ProxyExceptions NVARCHAR(max),
 CONSTRAINT FK_Browser_ApplicationID_Application_ApplicationID
FOREIGN KEY (ApplicationID) REFERENCES Application (ApplicationID),
 PRIMARY KEY (ApplicationID));
```

4.4. Table: BrowserTemplate

To allow working with application templates, application template tables are provided. They contain the same fields as the application tables.

ColumnName	DataType	Key	Not Null	DefaultValue	Comment
ApplicationTemplateID	INTEGER	PK, FK	NN		Unique application template ID (foreign key refers to table ApplicationTemplate)
StartPage	NVARCHAR (255)				Home page
CallPage	NVARCHAR (255)				Called at browser start
Proxy	NVARCHAR (255)				Proxy server (IP address or name)
ProxyType	INTEGER				Use proxy server **0|1**
BrowserType	INTEGER				**3** Firefox **4** Chromium

ColumnName	DataType	Key	Not Null	DefaultValue	Comment
KioskMode	SMALLINT(6)				Kiosk mode **0|1**
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				
ProxyExceptions	NVARCHAR (max)				Holds the entries of the browser proxy exception list

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationTemplateID

How to create

```
CREATE TABLE BrowserTemplate(
 ApplicationTemplateID INTEGER NOT NULL,
 StartPage NVARCHAR(255),
 CallPage NVARCHAR(255),
 Proxy NVARCHAR(255),
 ProxyType INTEGER,
 BrowserType INTEGER,
 KioskMode SMALLINT,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 ProxyExceptions NVARCHAR(max),
 CONSTRAINT FK_BrowserTemplate_ApplicationTemplateID_Applic-
ationTemplate_ApplicationTemplateID FOREIGN KEY (Applic-
ationTemplateID) REFERENCES ApplicationTemplate
(ApplicationTemplateID),
 PRIMARY KEY (ApplicationTemplateID));
```

4.5. Table: Emulation

Each entry describes an application of the type APPLTYPE_EMULATION.[1]

ColumnName	DataType	Key	Not Null	DefaultValue	Comment
ApplicationID	INTEGER	PK, FK	NN		Unique application ID (foreign key refers to table Application)
Type	NVARCHAR(255)				Emulation type **5250** 5250 terminal emulation **PowerTerm** PowerTerm session
Configuration	NVARCHAR(255)				**<configuration file>** Emulation types 9750, 3270, 5250
ConFile	NVARCHAR(255)				Connection file. Supported by emulation types 9750, 3270, 5250
ColorFile	NVARCHAR(255)				Color definition file. Supported by emulation types 9750, 3270, 5250
KeyboardFile	NVARCHAR(255)				Keyboard definition file. Supported by emulation types 9750, 3270, 5250
PTFile	NVARCHAR(255)				P-Key definition file. Supported by emulation types 9750, 3270, 5250

[1] From Scout 15 2110, only the emulation types 5250 and PowerTerm are supported.

ColumnName	DataType	Key	Not Null	DefaultValue	Comment
BTFile	NVARCHAR(255)				B-Key definition file. Supported by emulation types 9750, 3270, 5250
Server	NVARCHAR(255)				Host name of server to connect to (see HostIP attribute)
Application	NVARCHAR(255)				**<password>** Emulation type VNC
Backingstore	NVARCHAR(255)				**default|on|off**
DeadKeys	NVARCHAR(255)				**default|on|off**
Menubar	SMALLINT(6)				Show menu bar (supported by emulation type PowerTerm only)**0|1**
Toolbar	SMALLINT(6)				Show toolbar (supported by emulation type PowerTerm only) **0|1**
File1	NVARCHAR(max)				**<connection file content>** Emulation types 9750, 3270, 5250 **<pts file content>** Emulation type PowerTerm
File2	NVARCHAR(max)				**<configuration file content>** Emulation types 9750, 3270, 5250 **<ptc file content>** Emulation type PowerTerm
File3	NVARCHAR(max)				**<ptk file content>** Emulation type PowerTerm
File4	NVARCHAR(max)				**<ptp file content>** Emulation type PowerTerm
File5	NVARCHAR(max)				Unused
ReservedString1	NVARCHAR(255)				

ColumnName	DataType	Key	Not Null	DefaultValue	Comment
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationID

unicon

How to create

```
CREATE TABLE Emulation(
 ApplicationID INTEGER NOT NULL,
 Type NVARCHAR(255),
 Configuration NVARCHAR(255),
 ConFile NVARCHAR(255),
 ColorFile NVARCHAR(255),
 KeyboardFile NVARCHAR(255),
 PTFile NVARCHAR(255),
 BTFile NVARCHAR(255),
 Server NVARCHAR(255),
 Application NVARCHAR(255),
 Backingstore NVARCHAR (255),
 DeadKeys NVARCHAR (255),
 Menubar SMALLINT,
 Toolbar SMALLINT,
 File1 NVARCHAR(max),
 File2 NVARCHAR(max),
 File3 NVARCHAR(max),
 File4 NVARCHAR(max),
 File5 NVARCHAR(max),
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 CONSTRAINT FK_Emulation_ApplicationID_Application_ApplicationID FOREIGN KEY (ApplicationID) REFERENCES
Application (ApplicationID),
 PRIMARY KEY (ApplicationID))
```

4.6. Table: EmulationTemplate

To allow working with application templates, application template tables are provided. They contain the same fields as the application tables.[1]

ColumnName	DataType	Key	Not Null	DefaultValue	Comment
ApplicationTemplateID	INTEGER	PK, FK	NN		Unique application template ID (foreign key refers to table ApplicationTemplate)
Type	NVARCHAR(255)				Emulation type **5250** 5250 terminal emulation **PowerTerm** PowerTerm session
Configuration	NVARCHAR(255)				**<configuration file>** Emulation types 9750, 3270, 5250
ConFile	NVARCHAR(255)				Connection file. Supported by emulation types 9750, 3270, 5250
ColorFile	NVARCHAR(255)				Color definition file. Supported by emulation types 9750, 3270, 5250
KeyboardFile	NVARCHAR(255)				Keyboard definition file. Supported by emulation types 9750, 3270, 5250
PTFile	NVARCHAR(255)				P-Key definition file. Supported by emulation types 9750, 3270, 5250

[1] From Scout 15 2110, only the emulation types 5250 and PowerTerm are supported.

ColumnName	DataType	Key	Not Null	DefaultValue	Comment
BTFile	NVARCHAR(255)				B-Key definition file. Supported by emulation types 9750, 3270, 5250
Server	NVARCHAR(255)				Host name of server to connect to (see HostIP attribute)
Application	NVARCHAR(255)				**<password>** Emulation type VNC
Backingstore	NVARCHAR(255)				**default|on|off**
DeadKeys	NVARCHAR(255)				**default|on|off**
Menubar	SMALLINT(6)				Show menu bar (supported by emulation type PowerTerm only)**0|1**
Toolbar	SMALLINT(6)				Show toolbar (supported by emulation type PowerTerm only) **0|1**
File1	NVARCHAR(max)				**<connection file content>** Emulation types 9750, 3270, 5250 **<pts file content>** Emulation type PowerTerm
File2	NVARCHAR(max)				**<configuration file content>** Emulation types 9750, 3270, 5250 **<ptc file content>** Emulation type PowerTerm
File3	NVARCHAR(max)				**<ptk file content>** Emulation type PowerTerm
File4	NVARCHAR(max)				**<ptp file content>** Emulation type PowerTerm
File5	NVARCHAR(max)				Unused
ReservedString1	NVARCHAR(255)				

ColumnName	DataType	Key	Not Null	DefaultValue	Comment
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationTemplateID

unicon

How to create

```
CREATE TABLE EmulationTemplate(
 ApplicationTemplateID INTEGER NOT NULL,
 Type NVARCHAR(255),
 Configuration NVARCHAR(255),
 ConFile NVARCHAR(255),
 ColorFile NVARCHAR(255),
 KeyboardFile NVARCHAR(255),
 PTFile NVARCHAR(255),
 BTFile NVARCHAR(255),
 Server NVARCHAR(255),
 Application NVARCHAR(255),
 Backingstore NVARCHAR (255),
 DeadKeys NVARCHAR (255),
 Menubar SMALLINT,
 Toolbar SMALLINT,
 File1 NVARCHAR(max),
 File2 NVARCHAR(max),
 File3 NVARCHAR(max),
 File4 NVARCHAR(max),
 File5 NVARCHAR(max),
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 CONSTRAINT FK_EmulationTemplate_ApplicationTemplateID_ApplicationTemplate_ApplicationTemplateID FOREIGN
KEY (ApplicationTemplateID) REFERENCES ApplicationTemplate (ApplicationTemplateID),
 PRIMARY KEY (ApplicationTemplateID))
```

4.7. Table: ICA

Each entry describes an application of type APPLTYPE_ICA.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ApplicationID	INTEGER	PK, FK	NN		Unique application ID (foreign key referring to table Application)
Server	NVARCHAR(255)				Citrix server or Published Application name **<IP or name>** Citrix server **<published application name>** If a published application is specified with Application attribute
Application	NVARCHAR(255)				Application name **<program path>** **#<published application name>**
WorkDir	NVARCHAR(255)				Working directory
Username	NVARCHAR(255)				Username for logon
Pass	NVARCHAR(255)				User password for logon (encrypted) (use **000f4367616e78637f6c6e7e6e7a766760** for $ELUXPASSWORD single sign on)
Domain	NVARCHAR(255)				Domain name for logon
Autologin	SMALLINT(6)			0	Automatic logon **0|1**

ColumnName	DataType	Key	Not Null	Default Value	Comment
Resolution	INTEGER			7	Index to the following resolution list (ignored if ResolutionDefault is set):
AllowSmartCardLogon	SMALLINT(6)				Allow smart card login (attributes Username, Pass, Domain are ignored) 0|1
Width	NVARCHAR(255)			640	Screen width (depends on attribute Resolution) 640 if Resolution=0 800 if Resolution=1 1024 if Resolution=2 1280 if Resolution=3 1600 if Resolution=4 <width> if Resolution=5
Height	NVARCHAR(255)			480	Screen height (depends on attribute Resolution)
PercentOfScreen	NVARCHAR(255)			75	Percentage of screen size (ignored for Resolution<>6)
ColorDepth	INTEGER			2	Index to the following list (ignored if ColorDepthDefault is set): 0 4 bit (16 colors) 1 8 bit (256 colors) 2 16 bit (32 thousand colors) 3 24 bit (16 million colors)

ColumnName	DataType	Key	Not Null	Default Value	Comment
ColorMap	INTEGER				256 color mapping (ignored if ColorMapDefault is set) **1** private - exact colors **2** shared - approximate colors
Sound	INTEGER				Sound quality **0** sound disabled **1** low quality **2** medium quality **3** high quality
Crypt	INTEGER				Encryption level **0** Basic **1** 128 bit login only **2** 40 bit **3** 56 bit **4** 128 bit
Compress	INTEGER				Use Data compression **0|1**
BitmapCache	INTEGER				Use cache for bitmaps **0|1**

ColumnName	DataType	Key	Not Null	Default Value	Comment
ZLMouseMode	INTEGER				Mouse click feedback 0 Auto 1 On 2 Off
ZLKeyboardMode	INTEGER				Local text echo 0 Auto 1 On 2 Off
BrowserProtocol	INTEGER				Browser protocol 0 TCP/IP + HTTP 1 TCP/IP 2 SSL + HTTPS
ColorMapDefault	SMALLINT(6)				Use default color map 0|1
ColorDepthDefault	SMALLINT(6)				Use default color depth 0|1
ResolutionDefault	SMALLINT(6)				Use default resolution 0|1

ColumnName	DataType	Key	Not Null	Default Value	Comment
ServerLocationDefault	SMALLINT(6)				Use default server location **0|1**
ICAHost1	NVARCHAR(max)				IP of server location 1
ICAHost2	NVARCHAR(255)				IP of server location 2
ICAHost3	NVARCHAR(255)				IP of server location 3
ICAHost4	NVARCHAR(255)				IP of server location 4
ICAHost5	NVARCHAR(255)				IP of server location 5
EnableAudioInput	NVARCHAR(16)			No	Audio input enabled **Yes|No**
TransportReconnectDefault	NVARCHAR(16)				
TransportReconnectEnabled	NVARCHAR(16)			True	Enable reconnect **True|False**
TransportReconnectRetries	INTEGER			3	Reconnect retries
TransportReconnectDelay	INTEGER			30	Reconnect delay (in seconds)
ProxyUseDefault	NVARCHAR(16)				
ProxyType	NVARCHAR(255)			None	Type of proxy server **None|Auto|Socks|Secure**

ColumnName	DataType	Key	Not Null	Default Value	Comment
ProxyHost	NVARCHAR(255)				Connect via proxy server (ProxyType must be set) <proxy>:<port>
TcpGroupName1	NVARCHAR(255)				
TcpGroupName2	NVARCHAR(255)				
TcpGroupName3	NVARCHAR(255)				
MouseSendsCtrlV	NVARCHAR(16)			Off	
FileType	NVARCHAR(255)				
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationID

unicon

How to create

```
CREATE TABLE ICA(
 ApplicationID INTEGER NOT NULL,
 Server NVARCHAR(255),
 Application NVARCHAR(255),
 WorkDir NVARCHAR(255),
 Username NVARCHAR(255),
 Pass NVARCHAR(255),
 Domain NVARCHAR(255),
 Autologin SMALLINT,
 Resolution INTEGER,
 AllowSmartCardLogon SMALLINT,
 Width NVARCHAR(255),
 Height NVARCHAR(255),
 PercentOfScreen NVARCHAR(255),
 ColorDepth INTEGER,
 ColorMap INTEGER,
 Sound INTEGER,
 Crypt INTEGER,
 Compress INTEGER,
 BitmapCache INTEGER,
 ZLMouseMode INTEGER,
 ZLKeyboardMode INTEGER,
 BrowserProtocol INTEGER,
 ColorMapDefault SMALLINT,
 ColorDepthDefault SMALLINT,
 ResolutionDefault SMALLINT,
 ServerLocationDefault SMALLINT,
```

```
 ICAHost1 NVARCHAR(max),
 ICAHost2 NVARCHAR(255),
 ICAHost3 NVARCHAR(255),
 ICAHost4 NVARCHAR(255),
 ICAHost5 NVARCHAR(255),
 EnableAudioInput NVARCHAR(16),
 TransportReconnectDefault NVARCHAR(16),
 TransportReconnectEnabled NVARCHAR(16),
 TransportReconnectRetries INTEGER,
 TransportReconnectDelay INTEGER,
 ProxyUseDefault NVARCHAR(16),
 ProxyType NVARCHAR(255),
 ProxyHost NVARCHAR(255),
 TcpGroupName1 NVARCHAR(255),
 TcpGroupName2 NVARCHAR(255),
 TcpGroupName3 NVARCHAR(255),
 MouseSendsCtrlV NVARCHAR(16),
 FileType NVARCHAR(255),
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 CONSTRAINT FK_ICA_ApplicationID_Application_ApplicationID FOREIGN KEY (ApplicationID) REFERENCES Applic-
ation (ApplicationID),
 PRIMARY KEY (ApplicationID));
```

4.8. Table: ICATemplate

To allow working with application templates, application template tables are provided. They contain the same fields as the application tables.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ApplicationTemplateID	INTEGER	PK, FK	NN		Unique application template ID (foreign key refers to table ApplicationTemplate)
Server	NVARCHAR(255)				Citrix server or Published Application name **<IP or name>** Citrix server **<published application name>** If a published application is specified with Application attribute
Application	NVARCHAR(255)				Application name **<program path>** **#<published application name>**
WorkDir	NVARCHAR(255)				Working directory
Username	NVARCHAR(255)				Username for logon
Pass	NVARCHAR(255)				User password for logon (encrypted) (use **000f4367616e78637f6c6e7e6e7a766760** for $ELUXPASSWORD single sign on)
Domain	NVARCHAR(255)				Domain name for logon
Autologin	SMALLINT(6)			0	Automatic logon **0|1**

ColumnName	DataType	Key	Not Null	Default Value	Comment
Resolution	INTEGER			7	Index to the following resolution list (ignored if ResolutionDefault is set):
AllowSmartCardLogon	SMALLINT(6)				Allow smart card login (attributes Username, Pass, Domain are ignored) 0|1
Width	NVARCHAR(255)			640	Screen width (depends on attribute Resolution) 640 if Resolution=0 800 if Resolution=1 1024 if Resolution=2 1280 if Resolution=3 1600 if Resolution=4 <width> if Resolution=5
Height	NVARCHAR(255)			480	Screen height (depends on attribute Resolution)
PercentOfScreen	NVARCHAR(255)			75	Percentage of screen size (ignored for Resolution<>6)
ColorDepth	INTEGER			2	Index to the following list (ignored if ColorDepthDefault is set): 0 4 bit (16 colors) 1 8 bit (256 colors) 2 16 bit (32 thousand colors) 3 24 bit (16 million colors)

ColumnName	DataType	Key	Not Null	Default Value	Comment
ColorMap	INTEGER				256 color mapping (ignored if ColorMapDefault is set) 1 private - exact colors 2 shared - approximate colors
Sound	INTEGER				Sound quality 0 sound disabled 1 low quality 2 medium quality 3 high quality
Crypt	INTEGER				Encryption level 0 Basic 1 128 bit login only 2 40 bit 3 56 bit 4 128 bit
Compress	INTEGER				Use Data compression 0|1
BitmapCache	INTEGER				Use cache for bitmaps 0|1

ColumnName	DataType	Key	Not Null	Default Value	Comment
ZLMouseMode	INTEGER				Mouse click feedback 0 Auto 1 On 2 Off
ZLKeyboardMode	INTEGER				Local text echo 0 Auto 1 On 2 Off
BrowserProtocol	INTEGER				Browser protocol 0 TCP/IP + HTTP 1 TCP/IP 2 SSL + HTTPS
ColorMapDefault	SMALLINT(6)				Use default color map 0|1
ColorDepthDefault	SMALLINT(6)				Use default color depth 0|1
ResolutionDefault	SMALLINT(6)				Use default resolution 0|1

ColumnName	DataType	Key	Not Null	Default Value	Comment
ServerLocationDefault	SMALLINT(6)				Use default server location **0|1**
ICAHost1	NVARCHAR(max)				IP of server location 1
ICAHost2	NVARCHAR(255)				IP of server location 2
ICAHost3	NVARCHAR(255)				IP of server location 3
ICAHost4	NVARCHAR(255)				IP of server location 4
ICAHost5	NVARCHAR(255)				IP of server location 5
EnableAudioInput	NVARCHAR(16)			No	Audio input enabled **Yes|No**
TransportReconnectDefault	NVARCHAR(16)				
TransportReconnectEnabled	NVARCHAR(16)			True	Enable reconnect **True|False**
TransportReconnectRetries	INTEGER			3	Reconnect retries
TransportReconnectDelay	INTEGER			30	Reconnect delay (in seconds)
ProxyUseDefault	NVARCHAR(16)				
ProxyType	NVARCHAR(255)			None	Type of proxy server **None|Auto|Socks|Secure**

ColumnName	DataType	Key	Not Null	Default Value	Comment
ProxyHost	NVARCHAR(255)				Connect via proxy server (ProxyType must be set) <proxy>:<port>
TcpGroupName1	NVARCHAR(255)				
TcpGroupName2	NVARCHAR(255)				
TcpGroupName3	NVARCHAR(255)				
MouseSendsCtrlV	NVARCHAR(16)			Off	
FileType	NVARCHAR(255)				
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationTemplateID

unicon

How to create

```
CREATE TABLE ICATemplate(
 ApplicationTemplateID INTEGER NOT NULL,
 Server NVARCHAR(255),
 Application NVARCHAR(255),
 WorkDir NVARCHAR(255),
 Username NVARCHAR(255),
 Pass NVARCHAR(255),
 Domain NVARCHAR(255),
 Autologin SMALLINT,
 Resolution INTEGER,
 AllowSmartCardLogon SMALLINT,
 Width NVARCHAR(255),
 Height NVARCHAR(255),
 PercentOfScreen NVARCHAR(255),
 ColorDepth INTEGER,
 ColorMap INTEGER,
 Sound INTEGER,
 Crypt INTEGER,
 Compress INTEGER,
 BitmapCache INTEGER,
 ZLMouseMode INTEGER,
 ZLKeyboardMode INTEGER,
 BrowserProtocol INTEGER,
 ColorMapDefault SMALLINT,
 ColorDepthDefault SMALLINT,
 ResolutionDefault SMALLINT,
 ServerLocationDefault SMALLINT,
```

```
 ICAHost1 NVARCHAR(max),
 ICAHost2 NVARCHAR(255),
 ICAHost3 NVARCHAR(255),
 ICAHost4 NVARCHAR(255),
 ICAHost5 NVARCHAR(255),
 EnableAudioInput NVARCHAR(16),
 TransportReconnectDefault NVARCHAR(16),
 TransportReconnectEnabled NVARCHAR(16),
 TransportReconnectRetries INTEGER,
 TransportReconnectDelay INTEGER,
 ProxyUseDefault NVARCHAR(16),
 ProxyType NVARCHAR(255),
 ProxyHost NVARCHAR(255),
 TcpGroupName1 NVARCHAR(255),
 TcpGroupName2 NVARCHAR(255),
 TcpGroupName3 NVARCHAR(255),
 MouseSendsCtrlV NVARCHAR(16),
 FileType NVARCHAR(255),
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 CONSTRAINT FK_ICA_ApplicationTemplateID_ApplicationTemplate_ApplicationTemplateID FOREIGN KEY (Applic-
ationTemplateID) REFERENCES ApplicationTemplate (ApplicationTemplateID),
 PRIMARY KEY (ApplicationTemplateID));
```

4.9. Table: Local

Each entry describes an application of type APPLTYPE_LOCALSHELL.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ApplicationID	INTEGER	PK, FK	NN		Unique application ID (foreign key refers to table Application)
Type	NVARCHAR (255)				Local application type **CUSTOM** User-defined command **RESINFO** Resource info **SHELL** Local shell (XTERM) **SSH** Secure shell **QFM** File manager **QTT** Text editor **MPLAYER** Movie player **THUNDERBIRD** Thunderbird mail client **ICACC** ICA Connection Center **CLIENTINFO** Client information **MAGNIFIER** Magnifier
Parameter	NVARCHAR (255)				Command line parameter string For custom applications, this is the complete command string including all parameters
Hidden	SMALLINT(6)				Application is hidden to the user **0|1**
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationID

How to create

```
CREATE TABLE Local(
 ApplicationID INTEGER NOT NULL,
 Type NVARCHAR(255),
 Parameter NVARCHAR(255),
 Hidden SMALLINT,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 CONSTRAINT FK_LocalApplicationID_Application_ApplicationID FOREIGN
KEY (ApplicationID) REFERENCES Application (ApplicationID),
 PRIMARY KEY (ApplicationID));
```

4.10. Table: LocalTemplate

To allow working with application templates, application template tables are provided. They contain the same fields as the application tables.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ApplicationTemplateID	INTEGER	PK, FK	NN		Unique application template ID (foreign key refers to table ApplicationTemplate)

ColumnName	DataType	Key	Not Null	Default Value	Comment
Type	NVARCHAR (255)				Local application type **CUSTOM** User-defined command **RESINFO** Resource info **SHELL** Local shell (XTERM) **SSH** Secure shell **QFM** File manager **QTT** Text editor **MPLAYER** Movie player **THUNDERBIRD** Thunderbird mail client **ICACC** ICA Connection Center **CLIENTINFO** Client information **MAGNIFIER** Magnifier
Parameter	NVARCHAR (255)				Command line parameter string For custom applications, this is the complete command string including all parameters
Hidden	SMALLINT (6)				Application is hidden to the user **0|1**
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationTemplateID

How to create

```
CREATE TABLE LocalTemplate(
 ApplicationTemplateID INTEGER NOT NULL,
 Type NVARCHAR(255),
 Parameter NVARCHAR(255),
 Hidden SMALLINT,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 CONSTRAINT FK_LocalApplicationTemplateID_ApplicationTemplate_
ApplicationTemplateID FOREIGN KEY (ApplicationTemplateID)
REFERENCES ApplicationTemplate (ApplicationTemplateID),
 PRIMARY KEY (ApplicationTemplateID));
```

4.11. Table: PNA

Each entry describes an application of the type APPLTYPE_PNL.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ApplicationID	INTEGER	PK, FK	NN		Unique application ID (foreign key refers to table Application)
ServerUrl	NVARCHAR (255)				Server name or complete server URL (f.e. http://<server>/Citrix/PNAgent/config.xml)
Port	INTEGER			80	Server port
Username	NVARCHAR (255)				Username for logon
Pass	NVARCHAR (255)				User password for logon (encrypted) (use **000f4367616e78637f6c6e7e6e7a766760** for $ELUXPASSWORD single sign-on)
Domain	NVARCHAR (255)				Domain name
LogoffDelay	INTEGER				Logoff delay (in seconds), ignored if attribute UseLogoffDelay is not set

ColumnName	DataType	Key	Not Null	Default Value	Comment
UseLogoffDelay	SMALLINT (6)			0	Logoff after delay of LogoffDelay seconds **0|1**
AutostartFolder	NVARCHAR (255)				Folder name Applications located in this folder are started after logon
ShowLastUser	SMALLINT (6)			1	Last user is displayed in logon dialog **0|1**
AllowCancel	SMALLINT (6)			1	User can cancel logon dialog **0|1**
Resolution	INTEGER			0	Index to the following resolution list: **0** default **1** 640x480 **2** 800x600 **3** 1024x768 **4** 1280x1024 **5** 1600x1200 **6** seamless window **7** full screen
Colors	INTEGER			0	Index to the following list: **0** default **1** 4 bit (16 colors) **2** 8 bit (256 colors) **3** 16 bit (32 thousand colors) **4** 24 bit (16 million colors)
Audio	INTEGER			0	Sound quality **0** default **1** high quality **2** medium quality **3** low quality **4** sound disabled
ForcedLogoff	SMALLINT (6)			0	Logoff after last PN agent session was closed **0|1**

ColumnName	DataType	Key	Not Null	Default Value	Comment
LogoffSessions	SMALLINT (6)			0	Logoff PNA sessions if user logs off from client 0|1
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationID

How to create

```
CREATE TABLE PNA(
 ApplicationID INTEGER NOT NULL,
 ServerUrl NVARCHAR(255),
 Port INTEGER,
 Username NVARCHAR(255),
 Pass NVARCHAR(255),
 Domain NVARCHAR(255),
 LogoffDelay INTEGER,
 UseLogoffDelay SMALLINT,
 AutostartFolder NVARCHAR(255),
 ShowLastUser SMALLINT,
 AllowCancel SMALLINT,
 Resolution INTEGER,
 Colors INTEGER,
 Audio INTEGER,
 ForcedLogoff SMALLINT,
 LogoffSessions SMALLINT,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 CONSTRAINT FK_PNA_ApplicationID_Application_ApplicationID FOREIGN
KEY (ApplicationID) REFERENCES Application (ApplicationID),
 PRIMARY KEY (ApplicationID));
```

4.12. Table: PNATemplate

To allow working with application templates, application template tables are provided. They contain the same fields as the application tables.

ColumnName	DataType	Key	Not Nul-l	Defaul-t Value	Comment
Applic-ationTemplateID	INTEGER	P-K, FK	N-N		Unique application template ID (foreign key refers to table ApplicationTemplate)
ServerUrl	NVARCHA-R(255)				Server name or complete server URL (f.e. http://<server>/Citrix/PNAgent/config.x-ml)
Port	INTEGER			80	Server port
Username	NVARCHA-R(255)				Username for logon
Pass	NVARCHA-R(255)				User password for logon (encrypted) (use **000f4367616e78637f6c6e7e6e7a76676-0** for $ELUXPASSWORD single sign-on)
Domain	NVARCHA-R(255)				Domain name
LogoffDelay	INTEGER				Logoff delay (in seconds), ignored if attribute UseLogoffDelay is not set
UseLogoffDelay	SMALLINT (6)			0	Logoff after delay of LogoffDelay seconds **0|1**
AutostartFolder	NVARCHA-R(255)				Folder name Applications located in this folder are started after logon
ShowLastUser	SMALLINT (6)			1	Last user is displayed in logon dialog **0|1**
AllowCancel	SMALLINT (6)			1	User can cancel logon dialog **0|1**

ColumnName	DataType	Key	Not Nul-l	Defaul-t Value	Comment
Resolution	INTEGER			0	Index to the following resolution list: **0** default **1** 640x480 **2** 800x600 **3** 1024x768 **4** 1280x1024 **5** 1600x1200 **6** seamless window **7** full screen
Colors	INTEGER			0	Index to the following list: **0** default **1** 4 bit (16 colors) **2** 8 bit (256 colors) **3** 16 bit (32 thousand colors) **4** 24 bit (16 million colors)
Audio	INTEGER			0	Sound quality **0** default **1** high quality **2** medium quality **3** low quality **4** sound disabled
ForcedLogoff	SMALLINT (6)			0	Logoff after last PN agent session was closed **0|1**
LogoffSessions	SMALLINT (6)			0	Logoff PNA sessions if user logs off from client **0|1**
ReservedString1	NVARCHA-R(255)				
ReservedString2	NVARCHA-R(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationTemplateID

How to create

```
CREATE TABLE PNATemplate(
 ApplicationTemplateID INTEGER NOT NULL,
 ServerUrl NVARCHAR(255),
 Port INTEGER,
 Username NVARCHAR(255),
 Pass NVARCHAR(255),
 Domain NVARCHAR(255),
 LogoffDelay INTEGER,
 UseLogoffDelay SMALLINT,
 AutostartFolder NVARCHAR(255),
 ShowLastUser SMALLINT,
 AllowCancel SMALLINT,
 Resolution INTEGER,
 Colors INTEGER,
 Audio INTEGER,
 ForcedLogoff SMALLINT,
 LogoffSessions SMALLINT,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 CONSTRAINT FK_PNA_ApplicationTemplateID_ApplicationTemplate_Applic-
ationTemplateID FOREIGN KEY (ApplicationTemplateID) REFERENCES
ApplicationTemplate (ApplicationTemplateID)),
 PRIMARY KEY (ApplicationTemplateID ));
```

4.13. Table: RDP

Each entry describes an application of type APPLTYPE_RDP.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ApplicationID	INTEGER	PK, FK	NN		Unique application ID Foreign key refers to table Application
Server	NVARCHAR(255)				Server name (or IP)
Application	NVARCHAR(255)				Application to start
WorkDir	NVARCHAR(255)				Working directory of application
Username	NVARCHAR(255)				Username
Pass	NVARCHAR(255)				User password for logon (encrypted) (use **000f4367616e78637f6c6e7e6e7a766760** for $ELUXPASSWORD single sign on)
Domain	NVARCHAR(255)				Domain
RDPType	NVARCHAR(255)				**native**
Autologin	NVARCHAR(255)				Process automatic logon using **Username**, **Pass**, **Domain** **true|false**

ColumnName	DataType	Key	Not Null	Default Value	Comment
Resolution	SMALLINT(6)				Index to the following resolution list: **0** full screen **1** 640x480 **2** 800x600 **3** 1024x768 **4** 1152x864 **5** 1280x1024
Colors	SMALLINT(6)				Colors: **8 | 16 | 24**

ColumnName	DataType	Key	Not Null	Default Value	Comment
Keyboard	NVARCHAR(255)				Keyboard layout: "auto" auto "da" Dansk "de" German "de-ch" German(Switzerland) "en-gb" English(UK) "en-us" English(UK) "en-us_intl" English(US,international) "es" Spanish "fi" Finnish "fr" French "fr-be" French(Belgium) "fr-ch" French(Switzerland) "hu" Hungarian "is" Icelandic "it" Italian "nl" Dutch "nl-be" Dutch(Belgium) "no" Norwegian "pl" Polish "pt" Portuguese "ru" Russian "sk" Slovak "sl" Slovenian "sv" Swedish "tr" Turkish(Q) "trf" Turkish(F)

ColumnName	DataType	Key	Not Null	Default Value	Comment
Sound	SMALLINT(6)				Play sound: 0 sound off 3 play local 5 play remote
Printer	SMALLINT(6)				Local printer available in server session 0|1
Serial	SMALLINT(6)				Local serial ports available in server session 0|1
Parallel	SMALLINT(6)				Local parallel ports available in server session 0|1
Smartcard	SMALLINT(6)				Local smart card available in server session 0|1
Options	INTEGER			15	RDP options as a combination of the following values: 1 RDP_OPT_WMDECORATIONS Window manager decorations enabled 2 ~_ENCRYPTION Encryption enabled 4 ~_MOUSEMOTIONEVENTS Mouse motion events enabled 8 ~_NUMLOCKSYNC Numlock syncronisation enabled 16 ~_COMPRESSION Compression enabled

ColumnName	DataType	Key	Not Null	Default Value	Comment
Protocol	SMALLINT(6)			0	Protocol version: **0** auto **4** RDP V4 **5** RDP V5
Bandwidth	INTEGER			0	Bandwidth to use: **0** Standard **1** Modem 2 Broadband **3** LAN
DriveName1	NVARCHAR(255)				Drive letter **A - Z**
DriveLocalDir1	NVARCHAR(255)				Local directory to be mapped
DriveOption1	INTEGER				**1** RDP_DRIVEOPT_ON Drive enabled
DriveName2	NVARCHAR(255)				see DriveName1
DriveLocalDir2	NVARCHAR(255)				see DriveLocalDir1
DriveOption2	INTEGER				see DriveOption1
DriveName3	NVARCHAR(255)				see DriveName1
DriveLocalDir3	NVARCHAR(255)				see DriveLocalDir1
DriveOption3	INTEGER				see DriveOption1

ColumnName	DataType	Key	Not Null	Default Value	Comment
DriveName4	NVARCHAR(255)				see DriveName1
DriveLocalDir4	NVARCHAR(255)				see DriveLocalDir1
DriveOption4	INTEGER				see DriveOption1
DriveName5	NVARCHAR(255)				see DriveName1
DriveLocalDir5	NVARCHAR(255)				see DriveLocalDir1
DriveOption5	INTEGER				see DriveOption1
DriveName6	NVARCHAR(255)				see DriveName1
DriveLocalDir6	NVARCHAR(255)				see DriveLocalDir1
DriveOption6	INTEGER				see DriveOption1
DriveName7	NVARCHAR(255)				see DriveName1
DriveLocalDir7	NVARCHAR(255)				see DriveLocalDir1
DriveOption7	INTEGER				see DriveOption1
DriveName8	NVARCHAR(255)				see DriveName1
DriveLocalDir8	NVARCHAR(255)				see DriveLocalDir1
DriveOption8	INTEGER				see DriveOption1
DriveName9	NVARCHAR(255)				see DriveName1

ColumnName	DataType	Key	Not Null	Default Value	Comment
DriveLocalDir9	NVARCHAR(255)				see DriveLocalDir1
DriveOption9	INTEGER				see DriveOption1
DriveName10	NVARCHAR(255)				see DriveName1
DriveLocalDir10	NVARCHAR(255)				see DriveLocalDir1
DriveOption10	INTEGER				
ComName1	NVARCHAR(255)				
ComDev1	NVARCHAR(255)				
ComName2	NVARCHAR(255)				see ComName1
ComDev2	NVARCHAR(255)				see ComDev1
ComName3	NVARCHAR(255)				see ComName1
ComDev3	NVARCHAR(255)				see ComDev1
ComName4	NVARCHAR(255)				see ComName1
ComDev4	NVARCHAR(255)				see ComDev1
ComName5	NVARCHAR(255)				see ComName1
ComDev5	NVARCHAR(255)				see ComDev1
ComName6	NVARCHAR(255)				see ComName1

ColumnName	DataType	Key	Not Null	Default Value	Comment
ComDev6	NVARCHAR(255)				see ComDev1
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationID

HOW TO CREATE

```
CREATE TABLE RDP(
 ApplicationID INTEGER NOT NULL,
 Server NVARCHAR(255),
 Application NVARCHAR(255),
 WorkDir NVARCHAR(255),
 Username NVARCHAR(255),
 Pass NVARCHAR(255),
 Domain NVARCHAR(255),
 RDPType NVARCHAR(255),
 Autologin NVARCHAR(255),
 Resolution SMALLINT,
 Colors SMALLINT,
 Keyboard NVARCHAR(255),
 Sound SMALLINT,
 Printer SMALLINT,
 Serial SMALLINT,
 Parallel SMALLINT,
 Smartcard SMALLINT,
 Options INTEGER,
 Protocol SMALLINT,
 DriveName1 NVARCHAR(255),
 DriveLocalDir1 NVARCHAR(255),
 DriveOption1 INTEGER,
 DriveName2 NVARCHAR(255),
 DriveLocalDir2 NVARCHAR(255),
 DriveOption2 INTEGER,
 DriveName3 NVARCHAR(255),
```

```
 DriveLocalDir3 NVARCHAR(255),
 DriveOption3 INTEGER,
 DriveName4 NVARCHAR(255),
 DriveLocalDir4 NVARCHAR(255),
 DriveOption4 INTEGER,
 DriveName5 NVARCHAR(255),
 DriveLocalDir5 NVARCHAR(255),
 DriveOption5 INTEGER,
 DriveName6 NVARCHAR(255),
 DriveLocalDir6 NVARCHAR(255),
 DriveOption6 INTEGER,
 DriveName7 NVARCHAR(255),
 DriveLocalDir7 NVARCHAR(255),
 DriveOption7 INTEGER,
 DriveName8 NVARCHAR(255),
 DriveLocalDir8 NVARCHAR(255),
 DriveOption8 INTEGER,
 DriveName9 NVARCHAR(255),
 DriveLocalDir9 NVARCHAR(255),
 DriveOption9 INTEGER,
 DriveName10 NVARCHAR(255),
 DriveLocalDir10 NVARCHAR(255),
 DriveOption10 INTEGER,
 ComName1 NVARCHAR(255),
 ComDev1 NVARCHAR(255),
ComName2 NVARCHAR(255),
 ComDev2 NVARCHAR(255),
ComName3 NVARCHAR(255),
 ComDev3 NVARCHAR(255),
ComName4 NVARCHAR(255),
```

```
 ComDev4 NVARCHAR(255),
 ComName5 NVARCHAR(255),
  ComDev5 NVARCHAR(255),
 ComName6 NVARCHAR(255),
  ComDev6 NVARCHAR(255),
  ReservedString1 NVARCHAR(255),
  ReservedString2 NVARCHAR(255),
  ReservedInt1 INTEGER,
  ReservedInt2 INTEGER,
  CONSTRAINT FK_RDP_ApplicationID_Application_ApplicationID FOREIGN KEY (ApplicationID) REFERENCES Applic-
 ation (ApplicationID),
  PRIMARY KEY (ApplicationID));
```

4.14. Table: RDPTemplate

To allow working with application templates, application template tables are provided. They contain the same fields as the application tables.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ApplicationTemplateID	INTEGER	PK, FK	NN		Unique application template ID (foreign key refers to table ApplicationTemplate)
Server	NVARCHAR(255)				Server name (or IP)
Application	NVARCHAR(255)				Application to start
WorkDir	NVARCHAR(255)				Working directory of application
Username	NVARCHAR(255)				Username

ColumnName	DataType	Key	Not Null	Default Value	Comment
Pass	NVARCHAR(255)				User password for logon (encrypted) (use **000f4367616e78637f6c6e7e6e7a766760** for $ELUXPASSWORD single sign on)
Domain	NVARCHAR(255)				Domain
RDPType	NVARCHAR(255)				**native**
Autologin	NVARCHAR(255)				Process automatic logon using **Username**, **Pass**, **Domain** **true|false**
Resolution	SMALLINT(6)				Index to the following resolution list: **0** full screen **1** 640x480 **2** 800x600 **3** 1024x768 **4** 1152x864 **5** 1280x1024
Colors	SMALLINT(6)				Colors: **8 | 16 | 24**

ColumnName	DataType	Key	Not Null	Default Value	Comment
Keyboard	NVARCHAR(255)				Keyboard layout: "auto" auto "da" Dansk "de" German "de-ch" German(Switzerland) "en-gb" English(UK) "en-us" English(UK) "en-us_intl" English(US,international) "es" Spanish "fi" Finnish "fr" French "fr-be" French(Belgium) "fr-ch" French(Switzerland) "hu" Hungarian "is" Icelandic "it" Italian "nl" Dutch "nl-be" Dutch(Belgium) "no" Norwegian "pl" Polish "pt" Portuguese "ru" Russian "sk" Slovak "sl" Slovenian "sv" Swedish "tr" Turkish(Q) "trf" Turkish(F)

ColumnName	DataType	Key	Not Null	Default Value	Comment
Sound	SMALLINT(6)				Play sound: 0 sound off 3 play local 5 play remote
Printer	SMALLINT(6)				Local printer available in server session 0|1
Serial	SMALLINT(6)				Local serial ports available in server session 0|1
Parallel	SMALLINT(6)				Local parallel ports available in server session 0|1
Smartcard	SMALLINT(6)				Local smart card available in server session 0|1
Options	INTEGER			15	RDP options as a combination of the following values: 1 RDP_OPT_WMDECORATIONS Window manager decorations enabled 2 ~_ENCRYPTION Encryption enabled 4 ~_MOUSEMOTIONEVENTS Mouse motion events enabled 8 ~_NUMLOCKSYNC Numlock syncronisation enabled 16 ~_COMPRESSION Compression enabled

ColumnName	DataType	Key	Not Null	Default Value	Comment
Protocol	SMALLINT(6)			0	Protocol version: **0** auto **4** RDP V4 **5** RDP V5
Bandwidth	INTEGER			0	Bandwidth to use: **0** Standard **1** Modem 2 Broadband **3** LAN
DriveName1	NVARCHAR(255)				Drive letter **A** - **Z**
DriveLocalDir1	NVARCHAR(255)				Local directory to be mapped
DriveOption1	INTEGER				**1** RDP_DRIVEOPT_ON Drive enabled
DriveName2	NVARCHAR(255)				see DriveName1
DriveLocalDir2	NVARCHAR(255)				see DriveLocalDir1
DriveOption2	INTEGER				see DriveOption1
DriveName3	NVARCHAR(255)				see DriveName1
DriveLocalDir3	NVARCHAR(255)				see DriveLocalDir1
DriveOption3	INTEGER				see DriveOption1

ColumnName	DataType	Key	Not Null	Default Value	Comment
DriveName4	NVARCHAR(255)				see DriveName1
DriveLocalDir4	NVARCHAR(255)				see DriveLocalDir1
DriveOption4	INTEGER				see DriveOption1
DriveName5	NVARCHAR(255)				see DriveName1
DriveLocalDir5	NVARCHAR(255)				see DriveLocalDir1
DriveOption5	INTEGER				see DriveOption1
DriveName6	NVARCHAR(255)				see DriveName1
DriveLocalDir6	NVARCHAR(255)				see DriveLocalDir1
DriveOption6	INTEGER				see DriveOption1
DriveName7	NVARCHAR(255)				see DriveName1
DriveLocalDir7	NVARCHAR(255)				see DriveLocalDir1
DriveOption7	INTEGER				see DriveOption1
DriveName8	NVARCHAR(255)				see DriveName1
DriveLocalDir8	NVARCHAR(255)				see DriveLocalDir1
DriveOption8	INTEGER				see DriveOption1
DriveName9	NVARCHAR(255)				see DriveName1

ColumnName	DataType	Key	Not Null	Default Value	Comment
DriveLocalDir9	NVARCHAR(255)				see DriveLocalDir1
DriveOption9	INTEGER				see DriveOption1
DriveName10	NVARCHAR(255)				see DriveName1
DriveLocalDir10	NVARCHAR(255)				see DriveLocalDir1
DriveOption10	INTEGER				
ComName1	NVARCHAR(255)				
ComDev1	NVARCHAR(255)				
ComName2	NVARCHAR(255)				see ComName1
ComDev2	NVARCHAR(255)				see ComDev1
ComName3	NVARCHAR(255)				see ComName1
ComDev3	NVARCHAR(255)				see ComDev1
ComName4	NVARCHAR(255)				see ComName1
ComDev4	NVARCHAR(255)				see ComDev1
ComName5	NVARCHAR(255)				see ComName1
ComDev5	NVARCHAR(255)				see ComDev1
ComName6	NVARCHAR(255)				see ComName1

ColumnName	DataType	Key	Not Null	Default Value	Comment
ComDev6	NVARCHAR(255)				see ComDev1
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationTemplateID

HOW TO CREATE

```
CREATE TABLE RDPTemplate(
 ApplicationTemplateID INTEGER NOT NULL,
 Server NVARCHAR(255),
 Application NVARCHAR(255),
 WorkDir NVARCHAR(255),
 Username NVARCHAR(255),
 Pass NVARCHAR(255),
 Domain NVARCHAR(255),
 RDPType NVARCHAR(255),
 Autologin NVARCHAR(255),
 Resolution SMALLINT,
 Colors SMALLINT,
 Keyboard NVARCHAR(255),
 Sound SMALLINT,
 Printer SMALLINT,
 Serial SMALLINT,
 Parallel SMALLINT,
 Smartcard SMALLINT,
 Options INTEGER,
 Protocol SMALLINT,
 DriveName1 NVARCHAR(255),
 DriveLocalDir1 NVARCHAR(255),
 DriveOption1 INTEGER,
 DriveName2 NVARCHAR(255),
 DriveLocalDir2 NVARCHAR(255),
 DriveOption2 INTEGER,
 DriveName3 NVARCHAR(255),
```

```
 DriveLocalDir3 NVARCHAR(255),
 DriveOption3 INTEGER,
 DriveName4 NVARCHAR(255),
 DriveLocalDir4 NVARCHAR(255),
 DriveOption4 INTEGER,
 DriveName5 NVARCHAR(255),
 DriveLocalDir5 NVARCHAR(255),
 DriveOption5 INTEGER,
 DriveName6 NVARCHAR(255),
 DriveLocalDir6 NVARCHAR(255),
 DriveOption6 INTEGER,
 DriveName7 NVARCHAR(255),
 DriveLocalDir7 NVARCHAR(255),
 DriveOption7 INTEGER,
 DriveName8 NVARCHAR(255),
 DriveLocalDir8 NVARCHAR(255),
 DriveOption8 INTEGER,
 DriveName9 NVARCHAR(255),
 DriveLocalDir9 NVARCHAR(255),
 DriveOption9 INTEGER,
 DriveName10 NVARCHAR(255),
 DriveLocalDir10 NVARCHAR(255),
 DriveOption10 INTEGER,
 ComName1 NVARCHAR(255),
 ComDev1 NVARCHAR(255),
ComName2 NVARCHAR(255),
 ComDev2 NVARCHAR(255),
ComName3 NVARCHAR(255),
 ComDev3 NVARCHAR(255),
ComName4 NVARCHAR(255),
```

```
ComDev4 NVARCHAR(255),
ComName5 NVARCHAR(255),
 ComDev5 NVARCHAR(255),
ComName6 NVARCHAR(255),
 ComDev6 NVARCHAR(255),
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 CONSTRAINT FK_RDP_ApplicationTemplateID_ApplicationTemplate_ApplicationTemplateID FOREIGN KEY (Applic-
ationTemplateID) REFERENCES Application (ApplicationTemplateID),
 PRIMARY KEY (ApplicationTemplateID));
```

4.15. Table: SAP

Each entry describes an application of type APPLTYPE_SAPGUI.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ApplicationID	INTEGER	PK, FK	NN		Unique application ID (foreign key refers to table Application)
SapClassic	SMALLINT (6)				Use classic look and feel **0|1**
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Colums
PRIMARY	PRIMARY	ApplicationID

HOW TO CREATE

```
CREATE TABLE SAP(
 ApplicationID INTEGER NOT NULL,
 SapClassic SMALLINT,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 CONSTRAINT FK_SAP_ApplicationID_Application_ApplicationID FOREIGN
KEY (ApplicationID) REFERENCES Application (ApplicationID),
 PRIMARY KEY (ApplicationID));
```

4.16. Table: SAPTemplate

To allow working with application templates, application template tables are provided. They contain the same fields as the application tables.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ApplicationTemplateID	INTEGER	PK, FK	NN		Unique application template ID (foreign key refers to table ApplicationTemplate)
SapClassic	SMALLINT (6)				Use classic look and feel **0|1**
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Colums
PRIMARY	PRIMARY	ApplicationTemplateID

HOW TO CREATE

```
CREATE TABLE SAPTemplate(
 ApplicationTemplateID INTEGER NOT NULL,
 SapClassic SMALLINT,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 CONSTRAINT FK_BrowserTemplate_ApplicationTemplateID_Applic-
ationTemplate_ApplicationTemplateID FOREIGN KEY (Applic-
ationTemplateID) REFERENCES ApplicationTemplate
(ApplicationTemplateID),
 PRIMARY KEY (ApplicationTemplateID ));
```

4.17. Table: StoreFront

Each entry describes an application of type APPLTYPE_STOREFRONT.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ApplicationID	INTEGER	PK, FK	NN		Unique application ID (foreign key refers to table Application)
Stores	NVARCHAR (max)				Comma-separated list of URLs to Citrix stores
LogoffSessions	SMALLINT(6)			0	Logoff StoreFront sessions when user logs off from client **0|1**
DesktopsUseFullscreen	SMALLINT(6)			1	
AutostartResources	NVARCHAR (255)				Comma-separated list of resources to be started automatically
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				
Username	NVARCHAR (255)				
Pass	NVARCHAR (255)				
Domain	NVARCHAR (255)				
LogoffDelay	INT				Time to wait until the client logs off
UseLogoffDelay	SMALLINT				Flag indicating whether `LogoffDelay`is used
ForcedLogoff	SMALLINT				Flag indicating mode of logoff (wait for all session apps to be closed or immediately)

ColumnName	DataType	Key	Not Null	Default Value	Comment
ShowLastUser	SMALLINT				Flag indicating if last user credentials are to be be shown at logon
UseCRFile	BIT NOT NULL			0	Use a Citrix Provisioning file (.cr) instead
CRFile	NVARCHAR (255)				Provisioning file

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationID

How to create

```
CREATE TABLE Storefront(
ApplicationID INTEGER NOT NULL,
Stores NVARCHAR(max),
LogoffSessions SMALLINT,
DesktopsUseFullscreen SMALLINT,
Username NVARCHAR(255),
Pass NVARCHAR(255),
Domain NVARCHAR(255),
LogoffDelay INTEGER,
UseLogoffDelay SMALLINT,
ForcedLogoff SMALLINT,
ShowLastUser SMALLINT,
UseCRFile BIT NOT NULL,
CRFile NVARCHAR(255),
ReservedString1 NVARCHAR(255),
ReservedString2 NVARCHAR(255),
ReservedInt1 INTEGER,
ReservedInt2 INTEGER,
CONSTRAINT FK_StoreFront_ApplicationID_Application_ApplicationID
FOREIGN KEY (ApplicationID) REFERENCES Application (ApplicationID),
PRIMARY KEY (ApplicationID));
```

4.18. Table: StoreFrontTemplate

To allow working with application templates, application template tables are provided. They contain the same fields as the application tables.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ApplicationTemplateID	INTEGER	PK, FK	NN		Unique application template ID (foreign key refers to table ApplicationTemplate)
Stores	NVARCHAR (max)				Comma-separated list of URLs to Citrix stores
LogoffSessions	SMALLINT (6)			0	Logoff StoreFront sessions when user logs off from client **0|1**
DesktopsUseFullscreen	SMALLINT (6)			1	
AutostartResources	NVARCHAR (255)				Comma-separated list of resources to be started automatically
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				
Username	NVARCHAR (255)				
Pass	NVARCHAR (255)				
Domain	NVARCHAR (255)				
LogoffDelay	INT				Time to wait until the client logs off
UseLogoffDelay	SMALLINT				Flag indicating whether `LogoffDelay` is used
ForcedLogoff	SMALLINT				Flag indicating mode of logoff (wait for all session apps to be closed or immediately)
ShowLastUser	SMALLINT				Flag indicating whether last user credentials are shown on logon

ColumnName	DataType	Key	Not Null	Default Value	Comment
UseCRFile	BIT NOT NULL			0	Use a Citrix Provisioning file (.cr) instead
CRFile	NVARCHAR (255)				Provisioning file

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationTemplateID

How to create

```
CREATE TABLE StorefrontTemplate(
ApplicationTemplateID INTEGER NOT NULL,
Stores NVARCHAR(max),
LogoffSessions SMALLINT,
DesktopsUseFullscreen SMALLINT,
Username NVARCHAR(255),
Pass NVARCHAR(255),
Domain NVARCHAR(255),
LogoffDelay INTEGER,
UseLogoffDelay SMALLINT,
ForcedLogoff SMALLINT,
ShowLastUser SMALLINT,
UseCRFile BIT NOT NULL,
CRFile NVARCHAR(255),
ReservedString1 NVARCHAR(255),
ReservedString2 NVARCHAR(255),
ReservedInt1 INTEGER,
ReservedInt2 INTEGER,
CONSTRAINT FK_StoreFrontTemplate_ApplicationTemplateID_Applic-
ationTemplate_ApplicationTemplateID FOREIGN KEY (Applic-
ationTemplateID) REFERENCES ApplicationTemplate
(ApplicationTemplateID),
PRIMARY KEY (ApplicationTemplateID));
```

4.19. Table: VDA

Each entry describes an Virtual Desktop Application (VDA), type APPLTYPE_VD.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ApplicationID	INTEGER	PK, FK	NN		Unique application ID (foreign key refers to table Application)
VDType	INTEGER			1	Virtual desktop connection broker **1** LeoStream **2** VDM client **3** XenDesktop The advanced settings of a XenDesktop definition are stored as an entry with the same ApplicationID in the ICA table **4** VMware View The advanced settings of a VMware View definition are stored as an entry with the same ApplicationID in the RDP table
Server	NVARCHAR(255)				Server name or IP address
Username	NVARCHAR(255)				Username for logon
Pass	NVARCHAR(255)				User password for logon (encrypted) (use **000f4367616e78637f6c6e7e6e7a766760** for $ELUXPASSWORD single sign-on)

ColumnName	DataType	Key	Not Null	Default Value	Comment
Domain	NVARCHAR(255)				Domain name
UseSSL	SMALLINT(6)			0	Use secure connection via SSL **0|1**
ShowLastUser	SMALLINT(6)			0	Last user is displayed in logon dialog **0|1**
Options	INTEGER			0	Options (currently not used)
USBRules	NVARCHAR(255)				Defines the USB rules when connecting with VMware View
Protocol	NVARCHAR(255)			RDP	Defines the protocol which is used to present the virtual desktop **RDP | PCIOP | RGS | localvm**
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				
AllowH264	BIT		NN	0	for VMware Blast protocol
USBConnectOptions	INTEGER		NN	0	for VMware Blast protocol

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationID

HOW TO CREATE

```
CREATE TABLE VDA(
 ApplicationID INTEGER NOT NULL,
 VDType INTEGER,
 Server NVARCHAR(255),
 Username NVARCHAR(255),
 Pass NVARCHAR(255),
 Domain NVARCHAR(255),
 UseSSL SMALLINT,
 ShowLastUser SMALLINT,
 Options INTEGER,
USBRules NVARCHAR(255),
Protocol NVARCHAR(255),
ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 AllowH264 BIT NOT NULL,
 USBConnectOptions INTEGER NOT NULL,
CONSTRAINT FK_VDA_ApplicationID_Application_ApplicationID FOREIGN KEY (ApplicationID) REFERENCES Applic-
ation (ApplicationID),
 PRIMARY KEY (ApplicationID));
```

4.20. Table: VDATemplate

To allow working with application templates, application template tables are provided. They contain the same fields as the application tables.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ApplicationTemplateID	INTEGER	PK, FK	NN		Unique application template ID (foreign key refers to table ApplicationTemplate)
VDType	INTEGER			1	Virtual desktop connection broker **1** LeoStream **2** VDM client **3** XenDesktop The advanced settings of a XenDesktop definition are stored as an entry with the same ApplicationID in the ICA table **4** VMware View The advanced settings of a VMware View definition are stored as an entry with the same ApplicationID in the RDP table
Server	NVARCHAR(255)				Server name or IP address
Username	NVARCHAR(255)				Username for logon
Pass	NVARCHAR(255)				User password for logon (encrypted) (use **000f4367616e78637f6c6e7e6e7a766760** for $ELUXPASSWORD single sign-on)

ColumnName	DataType	Key	Not Null	Default Value	Comment
Domain	NVARCHAR(255)				Domain name
UseSSL	SMALLINT(6)			0	Use secure connection via SSL **0|1**
ShowLastUser	SMALLINT(6)			0	Last user is displayed in logon dialog **0|1**
Options	INTEGER			0	Options (currently not used)
USBRules	NVARCHAR(255)				Defines the USB rules when connecting with VMware View
Protocol	NVARCHAR(255)			RDP	Defines the protocol which is used to present the virtual desktop **RDP | PCIOP | RGS | localvm**
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				
AllowH264	BIT		NN	0	for VMware Blast protocol
USBConnectOptions	INTEGER		NN	0	for VMware Blast protocol

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ApplicationTemplateID

HOW TO CREATE

```
CREATE TABLE VDATemplate(
 ApplicationTemplateID INTEGER NOT NULL,
 VDType INTEGER,
 Server NVARCHAR(255),
 Username NVARCHAR(255),
 Pass NVARCHAR(255),
 Domain NVARCHAR(255),
 UseSSL SMALLINT,
 ShowLastUser SMALLINT,
 Options INTEGER,
 USBRules NVARCHAR(255),
 Protocol NVARCHAR(255),
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 AllowH264 BIT NOT NULL,
 USBConnectOptions INTEGER NOT NULL,
CONSTRAINT FK_VDA_ApplicationTemplateID_ApplicationTemplate_ApplicationTemplateID FOREIGN KEY (Applic-
ationTemplateID) REFERENCES ApplicationTemplate (ApplicationTemplateID),
 PRIMARY KEY (ApplicationTemplateID));
```

5. Setup (Device configuration)

5.1. Table: AuthDomain

Holds information about the friendly domain names.

ColumnName	DataType	Key	Not Null	Default Value	Comment
AuthDomainID	INTEGER IDENTITY (1,1)	PK	NN		Unique domain ID
SetupID	INTEGER		NN		
Name	NVARCHAR (255)		NN		
ServerList	NVARCHAR (255)		NN		Server, comma-separated server list or domain name
SearchBase	NVARCHAR (255)				Optional search base

IndexName	IndexType	Columns
PRIMARY	PRIMARY	HostListID

HOW TO CREATE

```
CREATE TABLE HostList(
AuthDomainID INTEGER IDENTITY (1,1) NOT NULL,
SetupID INTEGER NOT NULL,
Name NVARCHAR(255) NOT NULL,
ServerList NVARCHAR(255) NOT NULL,
SearchBase NVARCHAR(255),
CONSTRAINT PK_AuthDomain
PRIMARY KEY (AuthDomainID));
```

HOW TO INITIALIZE

```
INSERT INTO AuthDomain (Name, ServerList, SearchBase, SetupID)
SELECT 'DefaultDomain', AuthServer, AuthBase, SetupID
FROM Setup
WHERE AuthType=1 OR AuthType=3 OR AuthType=5;
```

5.2. Table: ClientConfig

This table holds all used client config data.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ClientConfigID	UNIQUEIDENTIFIER	PK	NN		
ClientData	VARBINARY(max)				
Created	DATETIME		NN		Timestamp of config creation

IndexName	IndexType	Columns
PK_ClientConfig	PRIMARY	ClientConfigID

HOW TO CREATE

```
CREATE TABLE ClientConfig( ClientConfigID UNIQUEIDENTIFIER NOT NULL
default NewSequentialID(), ClientData VARBINARY(max), Created
DATETIME NOT NULL, CONSTRAINT PK_ClientConfig PRIMARY KEY (Cli-
entConfigID))
```

5.3. Table: ComPort

Entriy defines a set of COM port parameters.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ComPortID	INTEGER	PK	NN		Unique ID of Com port setting
SetupID	INTEGER	FK	NN		Setup ID (foreign key refers to table Setup)
Speed	SMALLINT (6)				Com speed **0** 1200 **1** 2400 **2** 4800 **3** 9600 **4** 19200 **5** 38400 **6** 57600 **7** 115200
Parity	SMALLINT (6)				Com parity **0** None **1** Even **2** Odd
Stop	SMALLINT (6)				Com stop bits. **0** Stop bits: 1 **1** Stop bits: 2
Flow	SMALLINT (6)				Com flow control. **0** None **1** RTS/CTS **2** XON/XOFF **3** Both
Width	SMALLINT (6)				Com word width. **0** 5 Bits **1** 6 Bits **2** 7 Bits **3** 8 Bits
ReservedString1	NVARCHAR (255)				

ColumnName	DataType	Key	Not Null	Default Value	Comment
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ComPortID

HOW TO CREATE

```
CREATE TABLE ComPort(
 ComPortID INTEGER NOT NULL,
 SetupID INTEGER NOT NULL,
 Speed SMALLINT,
 Parity SMALLINT,
 Stop SMALLINT,
 Flow SMALLINT,
 Width SMALLINT,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 CONSTRAINT FK_ComPort_SetupID_Setup_SetupID FOREIGN KEY (SetupID)
REFERENCES Setup (SetupID),
 PRIMARY KEY (ComPortID));
```

5.4. Table: ConfigRun

This table is used to display the progress of a config run in the Scout Enterprise console.

ColumnName	DataType	Key	Not Null	Default Value	Comment
UpdateJobID	INTEGER	PK	NN		
DeviceCount	INTEGER				
DevicesDone	INTEGER				

IndexName	IndexType	Columns
PK_ConfigRun	PRIMARY	UpdateJobID

HOW TO CREATE

```
CREATE TABLE ConfigRun(UpdateJobID INTEGER NOT NULL, DeviceCount
INTEGER, DevicesDone INTEGER, CONSTRAINT PK_ConfigRun, PRIMARY KEY
(UpdateJobID))
```

5.5. Table: EluxVarList

Each entry describes an environment variable to be set on the client.

ColumnName	DataType	Key	Not Null	Default Value	Comment
EluxVarListID	INTEGER	PK	NN		Unique ID of variable entry
SetupID	INTEGER	FK			Setup ID (foreign key refers to table Setup)
VarKey	NVARCHAR (255)				User-specific environment variable (ELUX...)
VarValue	NVARCHAR (255)				ADS/LDAP variable
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	EluxVarListID

HOW TO CREATE

```
CREATE TABLE EluxVarList (
 EluxVarListID INTEGER NOT NULL,
 SetupID INTEGER,
 VarKey NVARCHAR(255),
 VarValue NVARCHAR(255),
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 FOREIGN KEY (SetupID) REFERENCES Setup (SetupID),
 PRIMARY KEY (EluxVarListID));
```

5.6. Table: Firmware

Each entry is referred from the Setup table and holds a set of parameters for firmware update purposes.

ColumnName	DataType	Key	Not Null	Default Value	Comment
FirmwareID	INTEGER	PK	NN		Unique ID of firmware settings
ServerID	INTEGER	FK			Server ID Foreign key refers to table Server.ServerID
ProxyPort	INTEGER				Port to connect proxy server
Protocol	SMALLINT (6)				Protocol used for firmware update: **0** HTTP **1** FTP HTTPS FTPS
ProxyType	SMALLINT (6)		NN	0	None | Static (Consumer) | Static (Provider) | Dynamic
Path	NVARCHAR (255)				Path where the image files (IDF) can be found
Username	NVARCHAR (255)				Username to logon to update server
Pass	NVARCHAR (255)				Encrypted password to logon to update server
ProxyServerID	INTEGER	FK	NN		Server for VPN connections
ServerForVPN	NVARCHAR (255)				Encrypted password to logon to update server
AutoCheckUpdate	SMALLINT (6)			0	Check for update on system start **0|1**
ConfirmUpdate	SMALLINT (6)			1	Update must be confirmed by user **0|1**

ColumnName	DataType	Key	Not Null	Default Value	Comment
CheckUpdateOnShutdown	SMALLINT (6)			0	Check for firmware update on client shutdown **0|1**
MaxDeferments	INTEGER			0	Number of possible deferments 0 to 3
DefermentTimes	NVARCHAR (255)			15	Selected deferment periods: 15, 30, 60, 120, 240, 480 Example: Enter **60, 120** to allow both deferment periods on the client
SecurityOptions	INTEGER				
BIOSFileName	NVARCHAR (255)				Name of `.udf` file for UEFI updates
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	FirmwareID

HOW TO CREATE

```
CREATE TABLE Firmware(
 FirmwareID INTEGER NOT NULL;
 ServerID INTEGER;
 ProxyPort INTEGER;
 ServerForVPN NVARCHAR(255);
 Protocol SMALLINT;
 ProxyType SMALLINT NOT NULL;
 Path NVARCHAR(255);
 Username NVARCHAR(255);
 Pass NVARCHAR(255);
 ProxyServerID INTEGER;
 AutoCheckUpdate SMALLINT NOT NULL;
 ConfirmUpdate SMALLINT NOT NULL;
 CheckUpdateOnShutdown SMALLINT;
 MaxDeferments INTEGER;
 DefermentTimes NVARCHAR(255);
 SecurityOptions INTEGER;
 ReservedString1 NVARCHAR(255);
 ReservedString2 NVARCHAR(255);
 ReservedInt1 INTEGER;
 ReservedInt2 INTEGER;
 BIOSFileName NVARCHAR(255);
 CONSTRAINT FK_Firmware_ServerID_Server_ServerID FOREIGN KEY
(ServerID) REFERENCES Server (ServerID);
 CONSTRAINT FK_Firmware_ProxyServerID_Server_ServerID FOREIGN KEY
(ProxyServerID) REFERENCES Server (ServerID);
 PRIMARY KEY (FirmwareID));
```

5.7. Table: Host

The table contains host information such as IP addresses and host names.

ColumnName	DataType	Key	Not Null	Default Value	Comment
HostID	INTEGER	PK	NN		Unique entry ID
IPAddress	NVARCHAR (255)		NN		IP address (dotted decimal notation)
Name	NVARCHAR (255)				Host name
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	HostID

HOW TO CREATE

```
CREATE TABLE Host(
 HostID INTEGER NOT NULL,
 IPAddress NVARCHAR(255) NOT NULL,
 Name NVARCHAR(255),
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (HostID));
```

5.8. Table: HostList

ColumnName	DataType	Key	Not Null	Default Value	Comment
HostListID	INTEGER	PK	NN		Unique host list entry ID
SetupID	INTEGER	FK	NN		Setup ID Foreign key refers to table Setup.SetupID
HostID	INTEGER	FK	NN		Entry ID Foreign key refers to table Host.HostID
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	HostListID

HOW TO CREATE

```
CREATE TABLE HostList(
 HostListID INTEGER NOT NULL,
 SetupID INTEGER NOT NULL,
 HostID INTEGER NOT NULL,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 CONSTRAINT FK_HostList_HostID_Host_HostID FOREIGN KEY (HostID)
REFERENCES Host (HostID),
 CONSTRAINT FK_HostList_SetupID_Setup_SetupID FOREIGN KEY (SetupID)
REFERENCES Setup (SetupID),
 PRIMARY KEY (HostListID));
```

5.9. Table: Language

This is a static table, do not change

A static table that contains all known locales of Scout Enterprise.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ColumnName	**DataType**	**Key**	**Not Null**	**Default Value**	**Comment**
LanguageID	INTEGER	PK	NN		Unique ID of language entry
Name	NVARCHAR (255)		NN		Locale such as **de_DE**, **en_US**)
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	LanguageID

HOW TO CREATE

```
CREATE TABLE Language(
 LanguageID INTEGER NOT NULL,
 Name NVARCHAR(255) NOT NULL,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (LanguageID));
```

How to initialize

```
INSERT INTO Language (LanguageID,Name) VALUES('1','da_DK');
 INSERT INTO Language (LanguageID,Name) VALUES('2','de_DE');
 INSERT INTO Language (LanguageID,Name) VALUES('3','de_CH');
 INSERT INTO Language (LanguageID,Name) VALUES('4','en_GB');
 INSERT INTO Language (LanguageID,Name) VALUES('5','en_US');
 INSERT INTO Language (LanguageID,Name) VALUES('6','fr_FR');
 INSERT INTO Language (LanguageID,Name) VALUES('7','fr_CH');
 INSERT INTO Language (LanguageID,Name) VALUES('8','fr_BE');
 INSERT INTO Language (LanguageID,Name) VALUES('9','nl_BE');
 INSERT INTO Language (LanguageID,Name) VALUES('10','it_IT');
 INSERT INTO Language (LanguageID,Name) VALUES('11','nl_NL');
 INSERT INTO Language (LanguageID,Name) VALUES('12','hr_HR');
 INSERT INTO Language (LanguageID,Name) VALUES('13','no_NO');
 INSERT INTO Language (LanguageID,Name) VALUES('14','pl_PL');
 INSERT INTO Language (LanguageID,Name) VALUES('15','pt_PT');
 INSERT INTO Language (LanguageID,Name) VALUES('16','ro_RO');
 INSERT INTO Language (LanguageID,Name) VALUES('17','sv_SE');
 INSERT INTO Language (LanguageID,Name) VALUES('18','sl_SI');
 INSERT INTO Language (LanguageID,Name) VALUES('19','es_ES');
 INSERT INTO Language (LanguageID,Name) VALUES('20','cs_CZ');
 INSERT INTO Language (LanguageID,Name) VALUES('21','hu_HU');
 INSERT INTO Language (LanguageID,Name) VALUES('22','tr_TR');
 INSERT INTO Language (LanguageID,Name) VALUES('23','ru_RU.ISO8859-
5');
 INSERT INTO Language (LanguageID,Name) VALUES('24','fi_FI');
 INSERT INTO Language (LanguageID,Name) VALUES('25','sk_SK');
 INSERT INTO Language (LanguageID,Name) VALUES('26','is_IS');
 INSERT INTO Language (LanguageID,Name) VALUES('27','et_EE');
 INSERT INTO Language (LanguageID,Name) VALUES('28','pt_BR');
 INSERT INTO Language (LanguageID,Name) VALUES('29','el_GR');
INSERT INTO Language (LanguageID,Name) VALUES('30','ja_JP');
```

5.10. Table: Multimedia

Holds the data from Setup > Multimedia.

ColumnName	DataType	Key	Not Null	Default Value	Comment
MultimediaID	INTEGER IDENTITY (1,1)	PK	NN		
MemberType	INTEGER		NN		Always 5 (means: Type is Setup)
MemberID	INTEGER		NN		Setup ID if MemberType is 5
OutputPriority	NVARCHAR (255)		NN		Comma-seperated list of output devices The first has precedence Example: digital,usb,analog
OutputUSB	INTEGER		NN		Volume level of usb output 0-100
OutputAnalog	INTEGER		NN		Volume level of analog output 0-100
OutputDigital	INTEGER		NN		Volume level of digital output 0-100
OutputMuteUSB	BIT		NN		Mute flag for usb output 0 | 1
OutputMuteAnalog	BIT		NN		Mute flag for analog output 0 | 1
OutputMuteDigital	BIT		NN		Mute flag for digital output 0 | 1
InputPriority	NVARCHAR (255)		NN		Comma seperated list of input devices The first has precedence Example: digital,usb,analog
InputUSB	INTEGER		NN		Level of usb input 0-100
InputAnalog	INTEGER		NN		Level of analog input 0-100
InputMuteUSB	BIT		NN		Mute flag for usb input **0** | **1**
InputMuteAnalog	BIT		NN		Mute flag for analog input **0** | **1**
SystemBeep	BIT		NN		Enable/disable system beep **0** | **1**

IndexName	IndexType	Columns
PRIMARY	PRIMARY	MultimediaID

HOW TO CREATE

```
CREATE TABLE Multimedia(
MultimediaID INTEGER IDENTITY (1,1) NOT NULL,
MemberType INTEGER NOT NULL,
MemberID INTEGER NOT NULL,
OutputPriority NVARCHAR(255) NOT NULL,
OutputUSB INTEGER NOT NULL,
OutputAnalog INTEGER NOT NULL,
OutputDigital INTEGER NOT NULL,
OutputMuteUSB BIT NOT NULL,
OutputMuteAnalog BIT NOT NULL,
OutputMuteDigital BIT NOT NULL,
InputPriority NVARCHAR(255) NOT NULL,
InputUSB INTEGER NOT NULL,
InputAnalog INTEGER NOT NULL,
InputMuteUSB BIT NOT NULL,
InputMuteAnalog BIT NOT NULL,
SystemBeep BIT NOT NULL,
CONSTRAINT PK_Multimedia
PRIMARY KEY (MultimediaID));
```

HOW TO INITIALIZE

```
INSERT INTO Multimedia SELECT
5,SetupID,'usb,analog,digital',MasterVolume/0.63,MasterVolume/0.63-
,MasterVolume/0.63,(MicMute & 2),(MicMute & 2),(MicMute &
2),'usb,analog',MicVolume,MicVolume,(MicMute & 1), (MicMute & 1),
XDMCPSound, SystemBeep FROM Setup WITH (HOLDLOCK TABLOCKX);
```

5.11. Table: NetDrive

Entry defines a network share and is connected to the **Setup** table.

ColumnName	DataType	Key	Not Null	Default Value	Comment
NetDriveID	INTEGER	PK	NN		Unique ID of net drive definition
SetupID	INTEGER	FK	NN		Setup ID Foreign key refers to table Setup.SetupID
Share	NVARCHAR (255)		NN		Share name (defined on server side)
Name	NVARCHAR (255)		NN		Name of mapped directory (absolute path) **/smb/**<Directory>
Directory	NVARCHAR (255)		NN		Name of mapped directory
ServerID	INTEGER	FK	NN		Server ID (foreign key refers to table Server.ServerID)
Username	NVARCHAR (255)		NN		Username
Pass	NVARCHAR (255)				Password (encrypted) (use **bQ@oEIOUuMSPUAXUTqEIX for** $ELUXPASSWORD single sign on)
UseKerberos	SMALLINT (6)				
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	NetDriveID

HOW TO CREATE

```
CREATE TABLE NetDrive(
 NetDriveID INTEGER NOT NULL,
 SetupID INTEGER NOT NULL,
 Share NVARCHAR(255) NOT NULL,
 Name NVARCHAR(255) NOT NULL,
 Directory NVARCHAR(255) NOT NULL,
 ServerID INTEGER NOT NULL,
 Username NVARCHAR(255) NOT NULL,
 Pass NVARCHAR(255),
 UseKerberos SMALLINT,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (NetDriveID),
CONSTRAINT FK_NetDrive_SetupID_Setup_SetupID FOREIGN KEY (SetupID)
REFERENCES Setup (SetupID),
 CONSTRAINT FK_NetDrive_ServerID_Server_ServerID FOREIGN KEY
(ServerID) REFERENCES Server(ServerID));
```

5.12. Table: NetworkProfile

ColumnName	DataType	Key	Not Null	Default Value	Comment
NetworkProfileID	INTEGER	PK	NN		Unique network profile ID
Name	NVARCHAR (255)				Network profile name
Protected	SMALLINT(6)			0	Network profile is read-only to the user **0|1**
NetworkType	INTEGER				4 Ethernet 5 WLAN 6 WWAN
AutoConnect	SMALLINT(6)				
SetupID	INTEGER	FK	NN		Setup ID (foreign key refers to table Setup.SetupID)
Speed	INTEGER				
Timeout	INTEGER		NN	240	Connection timeout
DHCPTimeout	INTEGER		NN	240	Connection timeout
BootMode	NVARCHAR (15)		NN		DHCP or off
IP_Address	NVARCHAR (15)				
Domain	NVARCHAR (255)				
Netmask	NVARCHAR (15)				
MTUSize	INTEGER				
Dot1x	SMALLINT(6)				
Dot1xAuthenticationRetries	INTEGER			4	
Dot1xAutoconnectRetries	INTEGER			3	
Dot1xOptional	BIT			0	
Dot1xTimeout					

ColumnName	DataType	Key	Not Null	Default Value	Comment
ConnectivityCheck	TINYINT			2	Internet connection test
MSN	NVARCHAR (255)				
WirelessChannel	NVARCHAR (10)				
WirelessPSK	NVARCHAR (255)				
WirelessSSID	NVARCHAR (255)				
HiddenSSID	BIT				
SeenSSIDS	NVARCHAR (4000)				List of BSSIDs for hidden WLANs, required for auto-connection
WirelessWPAMode	NVARCHAR (10)				
WirelessIdentity	NVARCHAR (255)				
WirelessWEPMode	INTEGER				
DialNumber	NVARCHAR (255)				
Username	NVARCHAR (255)				Username for Mobile internet profile
Pass	NVARCHAR (255)				Password for Mobile internet profile
Callback	SMALLINT				
UseIP	SMALLINT				Use Roaming for Mobile internet profile
APN	NVARCHAR (255)				APN for Mobile internet profile
UseLease	SMALLINT(6)			0	Use last DHCP lease if DHCP server is not available 0|1
ProfileOptions	INTEGER				

ColumnName	DataType	Key	Not Null	Default Value	Comment
ProxyType	SMALLINT				System-wide proxy: 0 none 1 manual 2 auto
ProxyURL	NVARCHAR (255)				Proxy server and port for manual proxy config, URL for automatic proxy config
ProxyExceptions	NVARCHAR (max)				List of network addresses that should not use the proxy, separated by semicolons
ProxyUsername	NVARCHAR (255)				
ProxyPassword	NVARCHAR (255)				
ProxyCredPassthrough	BIT				
VpnType	TINYINT				VPN client type 1 OpenVPN 2 Cisco AnyConnect 3 to 254 freely definable types
VpnConfig	NVARCHAR (4000)				Name of the OpenVPN configuration file which must exist on the client
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	NetworkProfileID

HOW TO CREATE

```
CREATE TABLE NetworkProfile(
NetworkProfileID INTEGER NOT NULL,
Name NVARCHAR(255),
Protected SMALLINT,
NetworkType INTEGER NOT NULL,
AutoConnect SMALLINT,
SetupID INTEGER NOT NULL,
Speed NVARCHAR(10),
Timeout INTEGER NOT NULL,
DHCPTimeout INTEGER NOT NULL,
BootMode NVARCHAR(15) NOT NULL,
IP_Address NVARCHAR(15),
Domain NVARCHAR(255),
Netmask NVARCHAR(15),
MTUSize INTEGER,
Dot1x SMALLINT,
Dot1xAuthenticationRetries INTEGER,
Dot1xAutoconnectRetries INTEGER
Dot1xOptional BIT,
Dot1xTimeout INTEGER,
ConnectivityCheck TINYINT,
MSN NVARCHAR(255),
WirelessChannel NVARCHAR(10),
WirelessPSK NVARCHAR(255),
WirelessSSID NVARCHAR(255),
HiddenSSID BIT,
SeenSSIDS NVARCHAR(4000),
WirelessWPAMode NVARCHAR(10),
WirelessIdentity NVARCHAR(255),
WirelessWEPMode INTEGER,
DialNumber NVARCHAR(255),
Username NVARCHAR(255),
Pass NVARCHAR(255),
Callback SMALLINT,
UseIP NVARCHAR(255),
APN NVARCHAR(255),
UseLease SMALLINT,
ProfileOptions INTEGER,
ProxyType SMALLINT,
ProxyURL NVARCHAR(255),
ProxyExceptions NVARCHAR(max),
ProxyUsername NVARCHAR(255),
ProxyPassword NVARCHAR(255),
ProxyCredPassthrough BIT,
```

```
VpnType TINYINT,
VpnConfig NVARCHAR(4000),
ReservedString1 NVARCHAR(255),
ReservedString2 NVARCHAR(255),
ReservedInt1 INTEGER,
ReservedInt2 INTEGER,
CONSTRAINT FK_NetworkProfile_SetupID_Setup_SetupID FOREIGN KEY
(SetupID) REFERENCES Setup (SetupID),
PRIMARY KEY (NetworkProfileID));
```

5.13. Table: PowerManagementProfile

Stores the powermanagement profiles

ColumnName	DataType	Key	Not Null	Default Value	Comment
PowerManagementProfileID	INTEGER IDENTITY (1,1)	PK	NN		Unique ID of the profile
SetupID	INTEGER		NN		Setup the profile belongs to
ProfileType	INTEGER		NN		Eco (Power saver) | High performance (Power)
BrightnessLevel	INTEGER		NN		
ActionOnLid	INTEGER		NN		Action when users close the lid
ActionOnPowerButton	INTEGER		NN		Action when users press the power button
ActionOnS3	INTEGER		NN		Action when device goes to sleep mode
ActionOnS3NoLogin	INTEGER		NN		Action when device goes to sleep mode and user is logged off
ActionOnSleepButton	INTEGER		NN	3	Keyboard Power/Sleep key can be configured to: 0 no action 2 shutdown 3 suspend
DPMS	BIT		NN		Switch off the display
DPMSDelay	INTEGER		NN		Delay for switching off display
S3	BIT		NN		Activate sleep mode (S3)
S3Delay	INTEGER		NN		Delay for activating S3
S3NoLogin	BIT		NN		

ColumnName	DataType	Key	Not Null	Default Value	Comment
NoLoginDelay	INTEGER		NN	600	Delay when user is logged off
ScreenSaver	BIT		NN		Activate screen saver
ScreenSaverDelay	INTEGER		NN		Delay for activating screen saver

IndexName	IndexType	Columns
PRIMARY	PRIMARY	PowerManagementProfileID

HOW TO CREATE

```
CREATE TABLE PowerManagementProfile(
PowerManagementProfileID INTEGER IDENTITY(1,1),
SetupID INTEGER NOT NULL,
ProfileType INTEGER NOT NULL,
BrightnessLevel INTEGER NOT NULL,
ActionOnLid INTEGER NOT NULL,
ActionOnPowerButton INTEGER NOT NULL,
ActionOnS3 INTEGER NOT NULL,
ActionOnS3NoLogin INTEGER NOT NULL,
ActionOnSleepButton INTEGER NOT NULL,
DPMS BIT NOT NULL,
DPMSDelay INTEGER NOT NULL,
S3 BIT NOT NULL,
S3Delay INTEGER NOT NULL,
ScreenSaver BIT NOT NULL,
ScreenSaverDelay INTEGER NOT NULL,
S3NoLogin BIT NOT NULL,
NoLoginDelay INTEGER NOT NULL DEFAULT 600,
CONSTRAINT PK_PowerManagementProfile
PRIMARY KEY (PowerManagementProfileID));
```

HOW TO INITIALIZE

```
INSERT INTO PowerManagementProfile (BrightnessLevel, ActionOnLid,
ActionOnPowerButton, DPMS, DPMSDelay, S3, S3Delay, ScreenSaver,
ScreenSaverDelay, setupid, ProfileType, ActionOnS3)
SELECT 100, 1, 2, PowerManagement, StandbyTime*60, 0, 600,
UseScreenSaver, XAutolockDelay *60, MemberID, 2, 3 from
```

```
screensettings where MemberType=5 AND (PowerManagement <> 1 OR
StandbyTime <> 10 OR UseScreenSaver <> 0 OR XAutolockDelay <> 3 );
```

5.14. Table: PredefinedCommand

Holds the data for predefined commands in Advanced options.

ColumnName	DataType	Key	Not Null	Default Value	Comment
PredefinedCommandID	INTEGER IDENTITY (1,1)	PK	NN		Unique ID of the command
Name	NVARCHAR (255)		NN		Name of the command
Command	NVARCHAR (255)		NN		Command string itself
RunAsSystem	BIT		NN		Flag indicates if command should be run with root credentials
Active	BIT		NN		Indicates whether a command is active
Administrators	NVARCHAR (255)				Allowed admins or admin groups.

IndexName	IndexType	Columns
PRIMARY	PRIMARY	PredefinedCommandID

HOW TO CREATE

```
CREATE TABLE PredefinedCommand(
PredefinedCommandID INTEGER IDENTITY (1,1) NOT NULL,
Name NVARCHAR(255) NOT NULL,
Command NVARCHAR(255) NOT NULL,
RunAsSystem BIT NOT NULL,
Active BIT NOT NULL,
Administrators NVARCHAR(max),
CONSTRAINT PK_PredefinedCommand
PRIMARY KEY (PredefinedCommandID));
```

5.15. Table: Printer

Entry is a printer definition and refers to an entry in the **Setup** table.

ColumnName	DataType	Key	Not Null	Default Value	Comment
PrinterID	INTEGER	PK	NN		Unique ID of printer definition
SetupID	INTEGER	FK	NN		Setup ID (foreign key refers to table Setup.SetupID)
Baud	INTEGER				Baud rate **1200|2400|4800|9600|19200|38400|57600|115200**
Name	NVARCHAR (255)		NN	lp	Local Printer name
Type	NVARCHAR (255)		NN	par	Printer type: **par** Printer on parallel port **ser** Printer on serial port **usb** Printer on USB port **net** Network printer
TextFilter	SMALLINT (6)		NN	0	Text filter is active **0|1**
PclFilter	NVARCHAR (255)			false	PCL filter is active **true|false**
Address	NVARCHAR (255)				Printer network name (or IP address)
Queue	NVARCHAR (255)				Queue name
Com	SMALLINT (6)			-1	COM port ID: **1** Printer on serial port 1
UseAsDefault	SMALLINT (6)			0	Default printer **0|1**
DriverName	NVARCHAR (255)				Driver name on server side used for auto-connected printer

ColumnName	DataType	Key	Not Null	Default Value	Comment
TpConnect	SMALLINT (6)			0	Use as ThinPrint printer **0|1**
TpClass	NVARCHAR (255)				ThinPrint class name
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	PrinterID

HOW TO CREATE

```
CREATE TABLE Printer(
 PrinterID INTEGER NOT NULL,
 SetupID INTEGER NOT NULL,
 Baud INTEGER,
 Name NVARCHAR(255) NOT NULL,
 Type NVARCHAR(255) NOT NULL,
 TextFilter SMALLINT NOT NULL,
 PclFilter NVARCHAR(255),
 Address NVARCHAR(255),
 Queue NVARCHAR(255),
 Com SMALLINT,
 UseAsDefault SMALLINT,
 DriverName NVARCHAR(255),
 TpConnect SMALLINT,
 TpClass NVARCHAR(255),
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (PrinterID),
 CONSTRAINT FK_Printer_SetupID_Setup_SetupID FOREIGN KEY (SetupID)
REFERENCES Setup(SetupID));
```

5.16. Table: Resolution

This is a static table, do not change

A static table that contains all known locales of Scout Enterprise.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ResolutionID	INTEGER IDENTITY (1,1)	PK	NN		Unique ID of language entry
Name	NVARCHAR (255)		NN		Name of resolution Example: **640x480** or **Auto**

IndexName	IndexType	Columns
PRIMARY	PRIMARY	LanguageID

HOW TO CREATE

```
CREATE TABLE Resolution
ResolutionID INTEGER IDENTITY (1,1) NOT NULL,
Name NVARCHAR(255) NOT NULL,
CONSTRAINT PK_Resolution
PRIMARY KEY (ResolutionID));
```

How to initialize

```
INSERT INTO Resolution (Name) VALUES ('Auto');
INSERT INTO Resolution (Name) VALUES ('640x480');
INSERT INTO Resolution (Name) VALUES ('800x600');
INSERT INTO Resolution (Name) VALUES ('1024x768');
INSERT INTO Resolution (Name) VALUES ('1152x864');
INSERT INTO Resolution (Name) VALUES ('1280x720');
INSERT INTO Resolution (Name) VALUES ('1280x800');
INSERT INTO Resolution (Name) VALUES ('1280x1024');
INSERT INTO Resolution (Name) VALUES ('1440x900');
INSERT INTO Resolution (Name) VALUES ('1600x1200');
INSERT INTO Resolution (Name) VALUES ('1680x1050');
INSERT INTO Resolution (Name) VALUES ('1920x1080');
INSERT INTO Resolution (Name) VALUES ('1920x1200');
INSERT INTO Resolution (Name) VALUES ('2048x1536');
INSERT INTO Resolution (Name) VALUES ('2560x1440');
INSERT INTO Resolution (Name) VALUES ('2560x1600');
INSERT INTO Resolution (Name) VALUES ('2560x2048');
INSERT INTO Resolution (Name) VALUES ('3840x2160');
INSERT INTO Resolution (Name) VALUES ('4096x2160');
```

5.17. Table: Screen

Entry defines the settings of a screen (monitor)

ColumnName	DataType	Key	Not Null	Default Value	Comment
ScreenID	INTEGER	PK	NN		Unique ID of screen
ScreenSettingsID	INTEGER			0	Reference to the parent screensettingsID
MonitorNumber	INTEGER			1	Monitor number
Name	NVARCHAR (255)				Name of monitor (automatically generated with Monitor_#monitornumber Exaple: **Monitor_1**
LayoutPos	INTEGER			0	
LayoutNext	INTEGER			0	
Options	INTEGER			6	Bit field **1**:Bit for primary **2**:Bit for activation **4**:Bit for UseDDC
Resolution	NVARCHAR (255)			#auto	Screen resolution **#auto|640x480** **|** **800x600** **|** **1024x768** **|1152x864|1280x1024|1600x1200**
Frequency	INTEGER			60	Screen frequency **60|70|75|80|85|90|100**
Rotation	INTEGER			0	Screen rotation **0|90|180|270**
Connection class	INTEGER			0	Monitor connection class - only necessary if the screen does NOT use DDC. **0** AUTO **1** DP (Display Port) **2** DVI **3** VGA

ColumnName	DataType	Key	Not Null	Default Value	Comment
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ScreenID

HOW TO CREATE

```
CREATE TABLE Screen(
ScreenID INTEGER NOT NULL,
ScreenSettingsID INTEGER,
MonitorNumber INTEGER,
Name NVARCHAR(255),
LayoutPos INTEGER,
LayoutNext INTEGER,
Options INTEGER,
Resolution NVARCHAR(255),
Frequency INTEGER,
Rotation INTEGER,
ConnectionClass INTEGER,
ReservedString1 NVARCHAR(255),
ReservedString2 NVARCHAR(255),
ReservedInt1 INTEGER,
ReservedInt2 INTEGER,
PRIMARY KEY (ScreenID));
```

5.18. Table: ScreenSaver

Screen saver settings belongs to a entry of table **Setup**.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ScreenSaverID	INTEGER	PK	NN		Unique ID of an entry
Mode	NVARCHAR(255)			blank	Screen saver mode: **blank** Blank screen **one** One single screen saver is selected **random** Random selection
Cycle	NVARCHAR(255)			0:05:00	Time after which the screen saver is restarted <hours:minutes:seconds>
PicDir	NVARCHAR(255)			/setup/xscreensaver	Client directory where pictures can be found by Slideshow
BannerText	NVARCHAR(255)			eLux NG	Literal text to be displayed by Fontglide
SelectedHack	NVARCHAR(255)			ripples	Currently selected screen saver (Mode must be set to **one**): **deco|fontglide|ripples|slideshow|sonar|substrate|xmatrix**
Deco	NVARCHAR(255)				Parameters for Deco <param1>=<value1>::<param2>=<value2>::<paramN>=<valueN>
Fontglide	NVARCHAR(255)				Parameters for Fontglide <param1>=<value1>::<param2>=<value2>::<paramN>=<valueN>

ColumnName	DataType	Key	Not Null	Default Value	Comment
Ripples	NVARCHAR(255)				Parameters for Ripples <param1>=<value1>::<param2>=<value2>::<paramN>=<valueN>
Slideshow	NVARCHAR(255)				Parameters for Slideshow <param1>=<value1>::<param2>=<value2>::<paramN>=<valueN>
Sonar	NVARCHAR(255)				Parameters for Sonar <param1>=<value1>::<param2>=<value2>::<paramN>=<valueN>
Substrate	NVARCHAR(255)				Parameters for Substrate <param1>=<value1>::<param2>=<value2>::<paramN>=<valueN>
XMatrix	NVARCHAR(255)				Parameters for XMatrix <param1>=<value1>::<param2>=<value2>::<paramN>=<valueN>
PopSquares	NVARCHAR(255)				Parameters for PopSquares <param1>=<value1>::<param2>=<value2>::<paramN>=<valueN>
SetupID	INTEGER	FK			Setup ID (foreign key refers to table Setup)
HTML5	NVARCHAR(255)				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ScreenSaverID

HOW TO CREATE

```
CREATE TABLE ScreenSaver(
 ScreenSaverID INTEGER NOT NULL,
 Mode NVARCHAR(255),
 Cycle NVARCHAR(255),
 PicDir NVARCHAR(255),
 BannerText NVARCHAR(255),
 SelectedHack NVARCHAR(255),
 Deco NVARCHAR(255),
 Fontglide NVARCHAR(255),
 Ripples NVARCHAR(255),
 Slideshow NVARCHAR(255),
 Sonar NVARCHAR(255),
 Substrate NVARCHAR(255),
 XMatrix NVARCHAR(255),
 SetupID INTEGER,
 HTML5 NVARCHAR(255),
 PRIMARY KEY (ScreenSaverID));
```

5.19. Table: ScreenSettings

ColumnName	DataType	Key	Not Null	Default Value	Comment
ScreenSettingsID	INTEGER	PK	NN	1	Unique ID of screen setting
MemberType	INTEGER			5 (TYPE_ SETUP)	5:The settings correspond to a setup entry 1:The settings correspond to an OU entry 4:The settings correspond to a device entry
MemberID	INTEGER			-1	Number of setup, OU or device entry
ScreenCount	INTEGER			0	Number of screens which should be defined
AdjustAutomatically	BIT			1	
Hotkeys	NVARCHAR (255)			0=<Ctrl><Alt>	Shortcut keys for hiding displays
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ScreenSettingsID

HOW TO CREATE

```
CREATE TABLE ScreenSettings(
ScreenSettingsID INTEGER,
MemberType INTEGER,
MemberID INTEGER,
ScreenCount INTEGER,
AdjustAutomatically BIT,
Hotkeys NVARCHAR(255) DEFAULT '0=<Ctrl><Alt>';
ReservedString1 NVARCHAR(255),
ReservedString2 NVARCHAR(255),
ReservedInt1 INTEGER,
ReservedInt2 INTEGER,
PRIMARY KEY (ScreenSettingsID));
```

HOW TO INITIALIZE

```
CREATE TABLE ScreenSettings(
ScreenSettingsID INTEGER,
```

5.20. Table: Server

ColumnName	DataType	Key	Not Null	Default Value	Comment
ServerID	INTEGER	PK	NN		Unique server ID
Description	NVARCHAR(255)				Arbitrary description
IPAddress	NVARCHAR(255)				Server IP address (dotted notation)
IPName	NVARCHAR(255)				Server host name
ServerTypeID	INTEGER		NN		Server type 1 SERVERTYPE_DNS Domain name server 2 SERVERTYPE_GW Router 3 SERVERTYPE_FONT Font server 4 SERVERTYPE_TIME Time server 5 SERVERTYPE_WOL WakeOnLAN server 6 SERVERTYPE_PROXY Proxy server 7 SERVERTYPE_FW Firmware server 8 SERVERTYPE_FILE File server
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ServerID

HOW TO CREATE

```
CREATE TABLE Server(
 ServerID INTEGER NOT NULL,
 Description NVARCHAR(255),
 IPAddress NVARCHAR(255),
 IPName NVARCHAR(255),
 ServerTypeID INTEGER NOT NULL,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (ServerID));
```

5.21. Table: Setup

This is one of the major tables. It stores device configuration data for OUs/groups or devices.

ColumnName	DataType	Key	Not Null	Default Value	Comment
SetupID	INTEGER	PK	NN		Unique setup ID
ShowTaskbar	SMALLINT (6)		NN		Show taskbar **0|1**
BackgroundImage	NVARCHAR (255)				File name of background image
TaskbarAlwaysOnTop	SMALLINT (6)		NN		Taskbar always visible on top **0|1**
TaskbarAutoHide	SMALLINT (6)		NN		Hide taskbar automatically **0|1**
TaskbarShowClock	SMALLINT (6)		NN		Show clock in taskbar **0|1**

ColumnName	DataType	Key	Not Null	Default Value	Comment
NextFocusKey	NVARCHAR (255)		NN	Up	Key component of keyboard shortcut to switch to next task **Up|Tab**
NextFocusMod	NVARCHAR (255)		NN	C1	Modifier component of keyboard shortcut to switch to next task **C1|1**
PrevFocusKey	NVARCHAR (255)		NN	Down	Key component of keyboard shortcut to switch to previous task **Down|Tab**
PrevFocusMod	NVARCHAR (255)		NN	C1	Modifier component of hotkey to switch to previous task **C1|S1**
DesktopBackColor	NVARCHAR (255)		NN	#668ab9	Desktop background color (RGB) #<RRGGBB>
LanguageID	INTEGER	FK	NN		Language ID (foreign key refers to table Language.LanguageID)

ColumnName	DataType	Key	Not Null	Default Value	Comment
TimeZoneID	INTEGER	FK	NN		Time zone ID (foreign key refers to table TimeZone.TimeZoneID)
Home	NVARCHAR (255)				Browser home directory
AllowX11Clients	SMALLINT (6)		NN		X11 clients allowed **0|1**
PointerProtocol	NVARCHAR (10)		NN	auto	Mouse pointer protocol **auto** auto-detect (Buttons=**5**) **PS/2** 2 button mouse (Buttons=**2**) or 3 button mouse (Buttons=**3**) **IMPS/2** wheel mouse (Buttons=**5**) **none** no mouse (Buttons=**none**)
LeftHandMouse	SMALLINT (6)		NN	0	Left hand mouse **0|1**
MultiClickTime	INTEGER		NN	500	Double click time 100(fast) - 900(slow)
DeadKeys	SMALLINT (6)		NN	1	Dead keys active **0|1**

ColumnName	DataType	Key	Not Null	Default Value	Comment
Numlock	SMALLINT (6)		NN	1	Numlock at start **0|1**
PointerSpeed	NVARCHAR (15)			2/1	Mouse acceleration 3/10(slow) - 13/2(fast)
KeyboardDelay	INTEGER			500	Keyboard delay (100-1510)
KeyboardSpeed	INTEGER		NN	30	Keyboard speed (10-250)
ConsoleSwitch	INTEGER			1	Enable switch to debug console **0|1**
XkbLayoutID	INTEGER	FK	NN	24	Keyboard Layout (foreign key refers to Xkblayout.XkbLayoutID)
BootMode	NVARCHAR (15)		NN		
Hostname	NVARCHAR (255)				
FirmwareID	INTEGER	FK	NN		Firmware ID Foreign key refers to Firmware.FirmwareID

ColumnName	DataType	Key	Not Null	Default Value	Comment
SmartCardDevice	NVARCHAR (15)		NN		
SmartCardCheck	NVARCHAR (15)		NN		
SmartCardDomain	NVARCHAR (255)				
RamDiskPercent	INTEGER		NN		
UseUSB	SMALLINT (6)				
AuthServer	NVARCHAR (255)				
AuthBase	NVARCHAR (255)				
AuthVersion	NVARCHAR (255)				

ColumnName	DataType	Key	Not Null	Default Value	Comment
AuthType	SMALLINT (6)			0	Type of user authentication **0** No **3** AD **5** AD+Smartcard **10** Evidian[1]
Mirror	SMALLINT (6)			1	Mirror enabled **0|1**
MirrorPassword	NVARCHAR (255)				
MirrorAcceptTimeout	INTEGER		NN	10	
port_lp	INTEGER			0	
port_usb	INTEGER			9101	
port_lpsave	INTEGER			9100	
ImageFile	NVARCHAR (255)				

[1]from Scout Enterprise 15.11

ColumnName	DataType	Key	Not Null	Default Value	Comment
LocalPassword	NVARCHAR (255)				
LockedFields	NVARCHAR (255)		NN		
ConnectTimeout	INTEGER				
DomainFieldStatus	INTEGER				
ShowLastUser	SMALLINT (6)			1	Last logged-on username displayed in user logon dialog **0|1**
LPDService	SMALLINT (6)			1	Enable LPD print service **0|1**
LogLevel	INTEGER			0	Verbosity level for diagnostic info: **1 | 6**
DiagURL	NVARCHAR (255)				URL to send diagnostic data to
ScreenRotate	INTEGER			0	Screen rotation **90|180|270**

ColumnName	DataType	Key	Not Null	Default Value	Comment
TpChannel	INTEGER			0	**0** TCP/IP **1** RDP/ICA
UseServerProfile	SMALLINT (6)			0	Use server stored user profiles **0|1**
ServerProfilePath	NVARCHAR (255)				Server UNC used to store user profiles
TaskbarOption	INTEGER			1843	Taskbar options as a combination of the following values: **1** QuickConfig keyboard **2** QuickConfig volume **4** QuickConfig peripherals **16** QuickConfig network **32** QuickConfig display **64** Show 'Show desktop' icon **256** (0x100) Show device information **512** (0x200) Show date and time **1024** (0x400) Show live information icons[1]

[1] from Scout Enterprise 15.7

ColumnName	DataType	Key	Not Null	Default Value	Comment
USBUserInfo	SMALLINT (6)			1	Show info message box if USB mass storage device is connected/disconnected **0|1**
IdleTimeout	INTEGER			10	After the client has established a connection to the server, the client will disconnect the TCP connection after the specified period of idle time

ColumnName	DataType	Key	Not Null	Default Value	Comment
DesktopOption	INTEGER			259	from Scout Enterprise 15.4 to 15.8
				771	from Scout Enterprise 15.9 The following options and their combinations are provided: **0** Do not show any icons on the desktop **1** Show only application icons (enabled by default) **2** Show volumes (enabled by default) **4** Show 'Home' folder **8** Show 'Computer' folder **16** Show 'Trash' folder **32** Show 'Network' folder **256** (0x100) Show Config panel icon (enabled by default) **512** (0x200) Sort configuration panel[1] (enabled by default)[2]
DesktopWriteable	SMALLINT (6)			0	Determines whether desktop objects can be modified **0|1**

[1]from Scout Enterprise 15.5
[2]from Scout Enterprise 15.9

ColumnName	DataType	Key	Not Null	Default Value	Comment
KeyboardModel	NVARCHAR (255)				Type of keyboard model **Auto** Keyboard model is detected automatically: configured by default **Cherry** Cherry G81-8000 keyboard **Desko** Desko MCx 4717 keyboard **K293** MFII keyboard **TYPE6** Sun Type6 keyboard **K235** Trimo 3270 keyboard **K257** Trimo 97801 keyboard
XPWriteFilter	INTEGER			1	Enables the write filter on XPe clients **0**|**1**
WMOption	INTEGER			1	Defines the window manager option **0** (no option)|**1** (Animated windows) |**2** (xinerama)
AuthOptions	INTEGER				**0** (no option) | **1** (AllowSCUserAuth)
ExtendedKeys	INTEGER				**0** No keys **127** All keys
KeepAlive	INTEGER				**60** Minimum **600** Default
NetworkOptions	INTEGER				**0** No options **1** Keep Alive is active

ColumnName	DataType	Key	Not Null	Default Value	Comment
ExtendedKeys	INTEGER				
MirrorAcceptTimeout	INTEGER		NN	10	Timeout in seconds for mirror request
LockControlPanel	BIT		NN	0	Protect control panel by password
PowerManagementProfileType	INTEGER		NN	0	Profile for power management
LogoffBeforeSuspend	BIT		NN	0	
WorkingHourFrom	INTEGER		NN	25200	
WorkingHourTo	INTEGER		NN	64800	
WorkingDays	INTEGER		NN	0	
Hotkeys	NVARCHAR (255)		NN	'1=<Ctrl><Alt>End'	Keyboard shortcut for screen saver
LowBatteryLevel	INTEGER		NN	10	Power management option
AutoRequestLicenses	INTEGER		NN	0	**0** Client does not request application licenses **1** Client requests application license if corresponding software package is installed **2** Client requests application license when corresponding software is started on the device
FollowMeDesktop	BIT		NN	1	

ColumnName	DataType	Key	Not Null	Default Value	Comment
ADAutologin	BIT				Auto-logon option for AD authentication
ADAutologinUser	NVARCHAR (255)				
ADAutologinPassword	NVARCHAR (255)				
ADAutologinDomain	NVARCHAR (255)				
TouchpadState	TINYINT			1	
PowerLineFrequency	TINYINT				Powerline frequency can be configured on the Hardware tab
SortOrder	TINYINT			1	Sort order of desktop icons (default is ascending)
UseSmartcard	BIT				Option allowing use of smart card for Evidian authentication
[Secret]	NVARCHAR (255)				Secret which must be configured (same as is used to set up the Evidian backend)
TimeFormat	TINYINT				Time format on devices **0** auto **1** 12 h time format **2** 24 h time format
ShutdownConfirmationTimerEnabled	BIT		NN	0	**0|1**

ColumnName	DataType	Key	Not Null	Default Value	Comment
ShutdownConfirmationTimer	TINYINT		NN	10	Time span in seconds (**10** to **60**)
KioskFile	NVARCHAR (max)				Kiosk mode settings defined via Console (Device config > Security)
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	SetupID

HOW TO CREATE

```
CREATE TABLE Setup(
 SetupID INTEGER NOT NULL,
 ShowTaskbar SMALLINT NOT NULL,
 BackgroundImage NVARCHAR (255),
 TaskbarAlwaysOnTop SMALLINT NOT NULL,
 TaskbarAutoHide SMALLINT NOT NULL,
 TaskbarShowClock SMALLINT NOT NULL,
 NextFocusKey NVARCHAR(255) NOT NULL,
 NextFocusMod NVARCHAR(255) NOT NULL,
 PrevFocusKey NVARCHAR(255) NOT NULL,
 PrevFocusMod NVARCHAR(255) NOT NULL,
 DesktopBackColor NVARCHAR(255) NOT NULL,
 LanguageID INTEGER NOT NULL,
 TimeZoneID INTEGER NOT NULL,
 Home NVARCHAR(255),
 AllowX11Clients SMALLINT NOT NULL,
 PointerProtocol NVARCHAR(10) NOT NULL,
 LeftHandMouse SMALLINT NOT NULL,
 MultiClickTime INTEGER NOT NULL,
 DeadKeys SMALLINT NOT NULL,
 Numlock SMALLINT NOT NULL,
 PointerSpeed NVARCHAR(15),
 KeyboardDelay INTEGER,
 KeyboardSpeed INTEGER NOT NULL,
 ConsoleSwitch INTEGER,
 XkbLayoutID INTEGER NOT NULL,
 Hostname NVARCHAR(255),
```

```
FirmwareID INTEGER NOT NULL,
SmartCardDevice NVARCHAR(15) NOT NULL,
SmartCardCheck NVARCHAR(15) NOT NULL,
SmartCardDomain NVARCHAR(255),
RamDiskPercent INTEGER NOT NULL,
UseUSB SMALLINT,
AuthServer NVARCHAR(255),
AuthBase NVARCHAR(255),
AuthVersion NVARCHAR(255),
AuthType SMALLINT,
Mirror SMALLINT,
MirrorPassword NVARCHAR(255),
MirrorAcceptTimeout INTEGER NOT NULL DEFAULT 10,
port_lp INTEGER,
port_usb INTEGER,
port_lpsave INTEGER,
ImageFile NVARCHAR(255),
LocalPassword NVARCHAR(255),
LockedFields NVARCHAR(255) NOT NULL,
ConnectTimeout INTEGER,
DomainFieldStatus INTEGER,
ShowLastUser SMALLINT,
LPDService SMALLINT,
LogLevel INTEGER,
DiagURL NVARCHAR(255),
ScreenRotate INTEGER,
TpChannel INTEGER,
UseServerProfile SMALLINT,
ServerProfilePath NVARCHAR(255),
```

```
TaskbarOption INTEGER,
USBUserInfo SMALLINT,
IdleTimeout INTEGER,
KeyboardModel NVARCHAR(255),
DesktopOption INTEGER,
DesktopWriteable SMALLINT,
XPWriteFilter INTEGER,
WMOption INTEGER,
AuthOptions INTEGER,
ExtendedKeys INTEGER,
KeepAlive INTEGER,
NetworkOptions INTEGER,
ExtendedKeys INTEGER,
MirrorAcceptTimeout INTEGER NOT NULL,
LockControlPanel BIT NOT NULL,
PowerManagementProfileType INTEGER NOT NULL,
LogoffBeforeSuspend BIT NOT NULL,
WorkingHourFrom INTEGER NOT NULL,
WorkingHourTo INTEGER NOT NULL,
WorkingDays INTEGER NOT NULL,
Hotkeys NVARCHAR(255) NOT NULL,
LowBatteryLevel INTEGER NOT NULL,
AutoRequestLicenses INTEGER NOT NULL DEFAULT 0,
FollowMeDesktop BIT NOT NULL DEFAULT 1,
ADAutologin BIT,
ADAutologinUser NVARCHAR(255),
ADAutologinPassword NVARCHAR(255),
ADAutologinDomain NVARCHAR(255),
TouchpadState TINYINT,
```

```
 PowerLineFrequency TINYINT,
 SortOrder TINYINT CONSTRAINT DF_Setup_SortOrder DEFAULT 1,
 UseSmartcard BIT,
 [Secret] __UNICON_VARCHAR__(255),
 TimeFormat TINYINT,
 ShutdownConfirmationTimerEnabled BIT NOT NULL DEFAULT 0;
 ShutdownConfirmationTimer TINYINT NOT NULL DEFAULT 10;
 KioskFile NVARCHAR(max)
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER, CONSTRAINT FK_Setup_FirmwareID_Firmware_FirmwareID FOREIGN KEY (FirmwareID)
REFERENCES Firmware (FirmwareID),
 CONSTRAINT FK_Setup_LanguageID_Language_LanguageID FOREIGN KEY (LanguageID) REFERENCES Language (Lan-
guageID),
 CONSTRAINT FK_Setup_TimeZoneID_TimeZone_TimeZoneID FOREIGN KEY (TimeZoneID) REFERENCES TimeZone
(TimeZoneID),
 CONSTRAINT FK_Setup_XkbLayoutID_XkbLayout_XkbLayoutID FOREIGN KEY (XkbLayoutID) REFERENCES XkbLayout
(XkbLayoutID),
 PRIMARY KEY (SetupID));
```

5.22. Table: ServerList

This is a helper table to normalize the relationship between **Setup** table and **Server** table.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ServerListID	INTEGER	PK	NN		Unique ID of server list entry
SetupID	INTEGER	FK	NN		Setup ID (foreign key refers to table Setup.SetupID)
ServerID	INTEGER	FK	NN		Server ID (foreign key refers to table Server.ServerID)
Instance	INTEGER			0	Determines whether the setup ID has to be interpreted as a network profile ID **0**: The setup ID is the real setup ID **1**: The setup ID has to be interpreted as a network profile ID
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
IndexName	**IndexType**	**Columns**
PRIMARY	PRIMARY	ServerListID

HOW TO CREATE

```
CREATE TABLE ServerList(
 ServerListID INTEGER NOT NULL,
 SetupID INTEGER NOT NULL,
 ServerID INTEGER NOT NULL,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 CONSTRAINT FK_ServerList_ServerID_Server_ServerID FOREIGN KEY
(ServerID) REFERENCES Server (ServerID),
PRIMARY KEY (ServerListID));
```

5.23. Table: TimeZone

Static table, do not change.

A static table that contains all known time zones of Scout Enterprise.

ColumnName	DataType	Key	Not Null	Default Value	Comment
TimeZoneID	INTEGER	PK	NN		Unique ID of time zone
Name	NVARCHAR (255)		NN		Zone name (e.g **Berlin GMT+1**, **London GMT**)
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	TimeZoneID

HOW TO CREATE

```
CREATE TABLE TimeZone(
 TimeZoneID INTEGER NOT NULL,
 Name NVARCHAR(255) NOT NULL,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (TimeZoneID));
```

HOW TO INITIALIZE

```
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(1,'America/Adak');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(2,'Asia/Aden');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(3,'Asia/Almaty');
```

```
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(4,'Asia/Amman');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(5,'Europe/Amsterdam');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(6,'Asia/Anadyr');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(7,'America/Anchorage');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(8,'Europe/Andorra');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(9,'America/Anguilla');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(10,'America/Antigua');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(11,'Asia/Aqtau');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(12,'Asia/Aqtobe');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(13,'America/Araguaina');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(14,'America/Aruba');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(15,'Asia/Ashgabat');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(16,'America/Asuncion');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(17,'Europe/Athens');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(18,'America/Atka');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(19,'Asia/Baghdad');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(20,'Asia/Bahrain');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(21,'Asia/Baku');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(22,'Asia/Bangkok');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(23,'America/Barbados');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(24,'Asia/Beirut');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(25,'America/Belem');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(26,'Europe/Belfast');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(27,'Europe/Belgrade');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(28,'America/Belize');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(29,'Europe/Berlin');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(30,'Asia/Bishkek');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(31,'America/Boa_
Vista');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(32,'America/Bogota');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(33,'America/Boise');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(34,'Europe/Bratislava');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(35,'Asia/Brunei');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
```

```
(36,'Europe/Brussels');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(37,'Europe/Bucharest');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(38,'Europe/Budapest');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(39,'America/Argentina/Buenos_Aires');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(40,'Asia/Calcutta');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(41,'America/Cambridge_Bay');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(42,'America/Cancun');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(43,'America/Caracas');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(44,'America/Argentina/Catamarca');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(45,'America/Cayenne');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(46,'America/Cayman');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(47,'America/Chicago');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(48,'America/Chihuahua');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(49,'Europe/Chisinau');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(50,'Asia/Choibalsan');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(51,'Asia/Chongqing');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(52,'Asia/Chungking');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(53,'Asia/Colombo');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(54,'Europe/Copenhagen');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(55,'America/Argentina/Cordoba');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(56,'America/Costa_
Rica');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(57,'America/Cuiaba');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(58,'America/Curacao');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(59,'Asia/Dacca');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(60,'Asia/Damascus');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
```

```
(61,'America/Danmarkshavn');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(62,'America/Dawson');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(63,'America/Dawson_
Creek');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(64,'America/Denver');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(65,'America/Detroit');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(66,'Asia/Dhaka');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(67,'Asia/Dili');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(68,'America/Dominica');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(69,'Asia/Dubai');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(70,'Europe/Dublin');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(71,'Asia/Dushanbe');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(72,'America/Edmonton');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(73,'America/Eirunepe');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(74,'America/El_Sal-
vador');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(75,'America/Ensenada');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(76,'America/Fort_
Wayne');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(77,'America/Fortaleza');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(78,'Asia/Gaza');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(79,'Europe/Gibraltar');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(80,'America/Glace_
Bay');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(81,'America/Godthab');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(82,'America/Goose_
Bay');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(83,'America/Grand_
Turk');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(84,'America/Grenada');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(85,'America/Guadeloupe');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(86,'America/Guatemala');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
```

```
(87,'America/Guayaquil');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(88,'America/Guyana');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(89,'America/Halifax');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(90,'Asia/Harbin');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(91,'America/Havana');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(92,'Europe/Helsinki');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(93,'America/Hermosillo');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(94,'Asia/Hong_
Kong');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(95,'Asia/Hovd');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(96,'America/Indiana/Indianapolis');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(97,'America/Inuvik');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(98,'America/Iqaluit');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(99,'Asia/Irkutsk');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(100,'Europe/Istanbul');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(101,'Asia/Jakarta');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(102,'America/Jamaica');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(103,'Asia/Jayapura');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(104,'Asia/Jerusalem');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(105,'America/Argentina/Jujuy');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(106,'America/Juneau');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(107,'Asia/Kabul');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(108,'Europe/Kaliningrad');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(109,'Asia/Kamchatka');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(110,'Asia/Karachi');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(111,'Asia/Kashgar');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(112,'Asia/Katmandu');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(113,'Europe/Kiev');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
```

```
(114,'America/Indiana/Knox');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(115,'America/Knox_
IN');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(116,'Asia/Krasnoyarsk');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(117,'Asia/Kuala_Lum-
pur');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(118,'Asia/Kuching');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(119,'Asia/Kuwait');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(120,'America/La_
Paz');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(121,'America/Lima');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(122,'Europe/Lisbon');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(123,'Europe/Ljubljana');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(124,'Europe/London');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(125,'America/Los_
Angeles');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(126,'America/Kentucky/Louisville');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(127,'Europe/Luxembourg');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(128,'Asia/Macao');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(129,'America/Maceio');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(130,'Europe/Madrid');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(131,'Asia/Magadan');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(132,'Europe/Malta');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(133,'America/Managua');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(134,'America/Manaus');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(135,'Asia/Manila');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(136,'America/Indiana/Marengo');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(137,'America/Martinique');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(138,'America/Mazatlan');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(139,'America/Argentina/Mendoza');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
```

```
(140,'America/Menominee');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(141,'America/Merida');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(142,'America/Mexico_
City');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(143,'Europe/Minsk');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(144,'America/Miquelon');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(145,'Europe/Monaco');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(146,'America/Monterrey');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(147,'America/Montevideo');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(148,'America/Kentucky/Monticello');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(149,'America/Montreal');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(150,'America/Montserrat');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(151,'Europe/Moscow');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(152,'Asia/Muscat');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(153,'America/Nassau');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(154,'America/New_
York');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(155,'Europe/Nicosia');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(156,'America/Nipigon');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(157,'America/Nome');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(158,'America/Noronha');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(159,'America/North_
Dakota/Center');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(160,'Asia/Novosibirsk');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(161,'Asia/Omsk');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(162,'Europe/Oslo');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(163,'America/Panama');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(164,'America/Pangnirtung');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(165,'America/Paramaribo');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(166,'Europe/Paris');
```

```
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(167,'Asia/Phnom_
Penh');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(168,'America/Phoenix');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(169,'Asia/Pontianak');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(170,'America/Port-
au-Prince');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(171,'America/Port_
of_Spain');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(172,'America/Porto_
Acre');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(173,'America/Porto_
Velho');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(174,'Europe/Prague');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(175,'America/Puerto_
Rico');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(176,'Asia/Pyongyang');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(177,'Asia/Qatar');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(178,'America/Rainy_
River');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(179,'Asia/Rangoon');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(180,'America/Rankin_
Inlet');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(181,'America/Recife');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(182,'America/Regina');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(183,'Europe/Riga');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(184,'America/Rio_
Branco');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(185,'Asia/Riyadh');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(186,'Europe/Rome');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(187,'America/Rosario');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(188,'Asia/Saigon');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(189,'Asia/Sakhalin');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(190,'Europe/Samara');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(191,'Asia/Samarkand');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(192,'Europe/San_
```

```
Marino');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(193,'America/Santiago');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(194,'America/Santo_
Domingo');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(195,'America/Sao_
Paulo');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(196,'Europe/Sarajevo');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(197,'America/Scoresbysund');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(198,'Asia/Seoul');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(199,'Asia/Shanghai');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(200,'America/Shiprock');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(201,'Europe/Simferopol');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(202,'Asia/Singapore');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(203,'Europe/Skopje');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(204,'Europe/Sofia');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(205,'America/St_
Johns');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(206,'America/St_
Kitts');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(207,'America/St_
Lucia');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(208,'America/St_Tho-
mas');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(209,'America/St_Vin-
cent');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(210,'Europe/Stockholm');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(211,'America/Swift_
Current');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(212,'Asia/Taipei');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(213,'Europe/Tallinn');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(214,'Asia/Tashkent');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(215,'Asia/Tbilisi');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(216,'America/Tegucigalpa');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(217,'Asia/Tehran');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(218,'Asia/Tel_
```

```
Aviv');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(219,'Asia/Thimbu');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(220,'Asia/Thimphu');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(221,'America/Thule');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(222,'America/Thunder_Bay');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(223,'America/Tijuana');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(224,'Europe/Tirane');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(225,'Europe/Tiraspol');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(226,'Asia/Tokyo');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(227,'America/Tortola');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(228,'Asia/Ujung_Pan-
dang');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(229,'Asia/Ulaanbaatar');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(230,'Asia/Ulan_
Bator');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(231,'Asia/Urumqi');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(232,'Europe/Uzhgorod');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(233,'Europe/Vaduz');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(234,'America/Vancouver');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(235,'Europe/Vatican');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(236,'America/Indiana/Vevay');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(237,'Europe/Vienna');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(238,'Asia/Vientiane');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(239,'Europe/Vilnius');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(240,'America/Virgin');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(241,'Asia/Vladivostok');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(242,'Europe/Warsaw');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
```

```
(243,'America/Whitehorse');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(244,'America/Winnipeg');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(245,'America/Yakutat');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(246,'Asia/Yakutsk');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(247,'Asia/Yekaterinburg');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(248,'America/Yellowknife');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(249,'Asia/Yerevan');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(250,'Europe/Zagreb');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(251,'Europe/Zaporozhye');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(252,'Europe/Zurich');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(253,'Europe/');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(254,'Europe/Volgograd');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(255,'Europe/Jersey');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(256,'Europe/Guernsey');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(257,'Europe/Isle_of_
Man');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(258,'Europe/Mariehamn');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(259,'Europe/Podgorica');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(260,'US/');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(261,'US/Pacific-
New');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(262,'US/Alaska');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(263,'US/Aleutian');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(264,'US/Arizona');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(265,'US/Central');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(266,'US/East-Indi-
ana');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(267,'US/Eastern');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(268,'US/Hawaii');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(269,'US/Indiana-
Starke');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(270,'US/Michigan');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(271,'US/Mountain');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(272,'US/Pacific');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(273,'US/Samoa');
```

```
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(274,'America/');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(275,'America/North_
Dakota/');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(276,'America/North_
Dakota/New_Salem');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(277,'America/North_
Dakota/Beulah');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(278,'America/Sitka');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(279,'America/Metlakatla');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(280,'America/Indiana/');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(281,'America/Indiana/Vincennes');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(282,'America/Indiana/Tell_City');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(283,'America/Indiana/Petersburg');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(284,'America/Indiana/Winamac');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(285,'America/Kentucky/');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(286,'America/Moncton');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(287,'America/Blanc-
Sablon');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(288,'America/Toronto');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(289,'America/Atikokan');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(290,'America/Resolute');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(291,'America/Matamoros');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(292,'America/Ojinaga');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(293,'America/Bahia_
Banderas');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(294,'America/Santa_
Isabel');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(295,'America/Argentina/');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(296,'America/Argentina/Salta');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
```

```
(297,'America/Argentina/Tucuman');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(298,'America/Argentina/La_Rioja');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(299,'America/Argentina/San_Juan');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(300,'America/Argentina/San_Luis');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(301,'America/Argentina/Rio_Gallegos');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(302,'America/Argentina/Ushuaia');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(303,'America/Argentina/ComodRivadavia');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(304,'America/Santarem');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(305,'America/Bahia');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(306,'America/Campo_
Grande');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(307,'America/St_
Barthelemy');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(308,'America/Marigot');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(309,'America/Lower_
Princes');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(310,'America/Kralendijk');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(311,'America/Buenos_
Aires');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(312,'America/Catamarca');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(313,'America/Coral_
Harbour');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(314,'America/Cordoba');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(315,'America/Indianapolis');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(316,'America/Jujuy');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(317,'America/Louisville');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(318,'America/Mendoza');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(319,'Asia/');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(320,'Asia/Macau');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(321,'Asia/Nicosia');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(322,'Asia/Kolkata');
```

```
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(323,'Asia/Makassar');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(324,'Asia/Qyzylorda');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(325,'Asia/Oral');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(326,'Asia/Kathmandu');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(327,'Asia/Hebron');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(328,'Asia/Ho_Chi_
Minh');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(329,'Asia/Novokuznetsk');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(330,'Asia/Riyadh87');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(331,'Asia/Riyadh88');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(332,'Asia/Riyadh89');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(333,'Asia/Istanbul');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(334,'Asia/Ashkhabad');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(335,'Africa/');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(336,'Africa/Algiers');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(337,'Africa/Luanda');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(338,'Africa/Porto-
Novo');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(339,'Africa/Gaborone');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(340,'Africa/Ouagadougou');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(341,'Africa/Bujumbura');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(342,'Africa/Douala');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(343,'Africa/Bangui');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(344,'Africa/Ndjamena');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(345,'Africa/Kinshasa');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(346,'Africa/Lubumbashi');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
```

```
(347,'Africa/Brazzaville');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(348,'Africa/Abidjan');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(349,'Africa/Djibouti');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(350,'Africa/Cairo');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(351,'Africa/Malabo');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(352,'Africa/Asmara');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(353,'Africa/Addis_
Ababa');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(354,'Africa/Libreville');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(355,'Africa/Banjul');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(356,'Africa/Accra');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(357,'Africa/Conakry');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(358,'Africa/Bissau');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(359,'Africa/Nairobi');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(360,'Africa/Maseru');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(361,'Africa/Monrovia');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(362,'Africa/Tripoli');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(363,'Africa/Blantyre');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(364,'Africa/Bamako');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(365,'Africa/Nouakchott');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(366,'Africa/Casablanca');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(367,'Africa/El_
Aaiun');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(368,'Africa/Maputo');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(369,'Africa/Windhoek');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(370,'Africa/Niamey');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(371,'Africa/Lagos');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
```

```
(372,'Africa/Kigali');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(373,'Africa/Sao_
Tome');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(374,'Africa/Dakar');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(375,'Africa/Freetown');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(376,'Africa/Mogadishu');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(377,'Africa/Johannesburg');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(378,'Africa/Khartoum');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(379,'Africa/Juba');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(380,'Africa/Mbabane');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(381,'Africa/Dar_es_
Salaam');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(382,'Africa/Lome');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(383,'Africa/Tunis');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(384,'Africa/Kampala');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(385,'Africa/Lusaka');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(386,'Africa/Harare');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(387,'Africa/Ceuta');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(388,'Africa/Asmera');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(389,'Africa/Timbuktu');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(390,'Atlantic/');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(391,'Atlantic/Cape_
Verde');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(392,'Atlantic/St_
Helena');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(393,'Atlantic/Faroe');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(394,'Atlantic/Reykjavik');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(395,'Atlantic/Azores');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(396,'Atlantic/Madeira');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(397,'Atlantic/Canary');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
```

```
(398,'Atlantic/Bermuda');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(399,'Atlantic/Stanley');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(400,'Atlantic/South_
Georgia');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(401,'Atlantic/Faeroe');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(402,'Atlantic/Jan_
Mayen');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(403,'Australia/');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(404,'Australia/Darwin');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(405,'Australia/Perth');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(406,'Australia/Eucla');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(407,'Australia/Brisbane');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(408,'Australia/Lindeman');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(409,'Australia/Adelaide');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(410,'Australia/Hobart');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(411,'Australia/Currie');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(412,'Australia/Melbourne');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(413,'Australia/Sydney');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(414,'Australia/Broken_Hill');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(415,'Australia/Lord_
Howe');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(416,'Australia/ACT');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(417,'Australia/Canberra');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(418,'Australia/LHI');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(419,'Australia/NSW');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(420,'Australia/North');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(421,'Australia/Queensland');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
```

```
(422,'Australia/South');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(423,'Australia/Tasmania');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(424,'Australia/Victoria');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(425,'Australia/West');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(426,'Australia/Yancowinna');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(427,'Indian/');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(428,'Indian/Comoro');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(429,'Indian/Antananarivo');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(430,'Indian/Mauritius');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(431,'Indian/Mayotte');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(432,'Indian/Reunion');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(433,'Indian/Mahe');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(434,'Indian/Kerguelen');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(435,'Indian/Chagos');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(436,'Indian/Maldives');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(437,'Indian/Christmas');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(438,'Indian/Cocos');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(439,'Pacific/');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(440,'Pacific/Rarotonga');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(441,'Pacific/Fiji');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(442,'Pacific/Gambier');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(443,'Pacific/Marquesas');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(444,'Pacific/Tahiti');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(445,'Pacific/Guam');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(446,'Pacific/Tarawa');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(447,'Pacific/Enderbury');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
```

```
(448,'Pacific/Kiritimati');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(449,'Pacific/Saipan');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(450,'Pacific/Majuro');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(451,'Pacific/Kwajalein');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(452,'Pacific/Chuuk');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(453,'Pacific/Pohnpei');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(454,'Pacific/Kosrae');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(455,'Pacific/Nauru');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(456,'Pacific/Noumea');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(457,'Pacific/Auckland');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(458,'Pacific/Chatham');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(459,'Pacific/Niue');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(460,'Pacific/Norfolk');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(461,'Pacific/Palau');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(462,'Pacific/Port_
Moresby');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(463,'Pacific/Pitcairn');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(464,'Pacific/Pago_
Pago');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(465,'Pacific/Apia');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(466,'Pacific/Guadalcanal');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(467,'Pacific/Fakaofo');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(468,'Pacific/Tongatapu');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(469,'Pacific/Funafuti');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(470,'Pacific/Johnston');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(471,'Pacific/Midway');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(472,'Pacific/Wake');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
```

```
(473,'Pacific/Efate');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(474,'Pacific/Wallis');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(475,'Pacific/Honolulu');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(476,'Pacific/Easter');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(477,'Pacific/Galapagos');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(478,'Pacific/Samoa');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(479,'Pacific/Yap');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(480,'Pacific/Truk');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(481,'Pacific/Ponape');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(482,'Etc/');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(483,'Etc/GMT');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(484,'Etc/UTC');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(485,'Etc/UCT');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(486,'Etc/GMT-14');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(487,'Etc/GMT-13');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(488,'Etc/GMT-12');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(489,'Etc/GMT-11');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(490,'Etc/GMT-10');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(491,'Etc/GMT-9');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(492,'Etc/GMT-8');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(493,'Etc/GMT-7');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(494,'Etc/GMT-6');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(495,'Etc/GMT-5');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(496,'Etc/GMT-4');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(497,'Etc/GMT-3');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(498,'Etc/GMT-2');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(499,'Etc/GMT-1');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(500,'Etc/GMT+1');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(501,'Etc/GMT+2');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(502,'Etc/GMT+3');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(503,'Etc/GMT+4');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(504,'Etc/GMT+5');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(505,'Etc/GMT+6');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(506,'Etc/GMT+7');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(507,'Etc/GMT+8');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(508,'Etc/GMT+9');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(509,'Etc/GMT+10');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(510,'Etc/GMT+11');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(511,'Etc/GMT+12');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
```

```
(512,'Etc/Universal');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(513,'Etc/Zulu');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES
(514,'Etc/Greenwich');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(515,'Etc/GMT-0');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(516,'Etc/GMT+0');
INSERT INTO TimeZone (TimeZoneID, Name) VALUES(517,'Etc/GMT0');
```

5.24. Table: XkbLayout

Static table, do not change.

Holds a list of available keyboard layout names.

ColumnName	DataType	Key	Not Null	Default Value	Comment
XkbLayoutID	INTEGER	PK	NN		Unique ID of keyboard layout entry
Name	NVARCHAR (255)		NN		Locale name (e.g. **de**, **us**)
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	XkbLayoutID

HOW TO CREATE

```
CREATE TABLE XkbLayout(
 XkbLayoutID INTEGER NOT NULL,
 Name NVARCHAR(255) NOT NULL,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (XkbLayoutID));
```

How to initialize

```
INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('1','be');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('2','cz');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('3','us_cz');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('4','dk');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('5','nl');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('6','gb');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('7','us');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('8','fr');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('9','fr_CH');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('10','de');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('11','de_CH');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('12','hu');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('13','it');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('14','no');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('15','pl');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('16','pl(pro-
grammers)');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('17','pt');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('18','sk');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('19','see1');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('20','es');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('21','se');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('22','trf');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('23','trq');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('24','en_US');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('25','be(nl)');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('26','ru');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('27','fi');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('28','is');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('29','ee');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('30','fo');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('31','jp');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('32','bg');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('33','la');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('34','br(abnt)');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('35','br(abnt2)');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('36','bg(phon-
etic)');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('37','el');
 INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('38','hr');
INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('39','jp
(OADG109A)');
INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('40','ca(fr)');
INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('41','ca(multix)');
```

```
INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('42','ca(fr-leg-
acy)');
INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('43','lv');
INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('44','lt');
INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('45','lt(std)');
INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('46','lt(ibm)');
INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('47','ie');
INSERT INTO XkbLayout (XkbLayoutID,Name) VALUES('48','pl(std)');
```

6. Entity Schedule

6.1. Table: ScheduledDevice

Bundles commands which are applied to multiple devices in the device list

ColumnName	DataType	Key	Not Null	Default Value	Comment
ScheduledDeviceID	INTEGER IDENTITY(1,1)	PK	NN		
ScheduleJobGUID	UNIQUEIDENTIFIER		NN		Each job is provided with a GUID
Mac	NVARCHAR(255)		NN		MAC address of the device the command belongs to
ClientIdentifier	UNIQUEIDENTIFIER		NN		Unique identifier for client devices and eLux portable sticks

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ScheduledDeviceID

```
HOW TO CREATE
CREATE TABLE ScheduledDevice(
ScheduledDeviceID INTEGER IDENTITY(1,1) NOT NULL,
ScheduleJobGUID UNIQUEIDENTIFIER,
Mac NVARCHAR(255) NOT NULL,
ClientIdentifier UNIQUEIDENTIFIER NOT NULL,,
CONSTRAINT PK_ScheduledDevice
PRIMARY KEY (ScheduledDeviceID));
```

6.2. Table: ScheduleJob

The scheduler of Scout Enterprise uses this table to store and manage all planned jobs.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ScheduleJobID	INTEGER	PK	NN		Unique ID of scheduled job
Type	INTEGER				Job type **1** SCD_TYPE_DEVICE ID refers to Device **2** SCD_TYPE_GROUP ID refers to Groups
Id	INTEGER	(FK)			ID of scheduled entity Depends on Type Refers to table Groups or Device
Addr1	NVARCHAR(255)				Address of entity

ColumnName	DataType	Key	Not Null	Default Value	Comment
Addr2	NVARCHAR(255)				
Occur	INTEGER				
Days	NVARCHAR(255)				
ScheduleHour	INTEGER				
ScheduleMin	INTEGER				
ScheduleTime	INTEGER				
Cmd	NVARCHAR(255)				
sParam	NVARCHAR(max)				
iParam	INTEGER				
ScheduleOption	INTEGER				
TimeCreated	INTEGER		NN		Creation time (Unix time)
TimeModified	INTEGER		NN		Last modification (Unix time)
TimeDone	INTEGER		NN		Time of completion (Unix time)
Status	INTEGER				Not used
SleepAfterSend	INTEGER				Time period in ms to go to sleep mode after sending commands to devices. This is useful when commands are sent to many devices

ColumnName	DataType	Key	Not Null	Default Value	Comment
Administrator	NVARCHAR(255)				Administrator who has scheduled the job
Result	INTEGER				Not used
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				
MessageTitle	NVARCHAR(255)				Message title for SEND MESSAGE commands
RootOU	INTEGER				Required to check if the scheduled jobs are within the admin's OU hierarchy
FriendlyName	NVARCHAR(255)				Friendly name for command

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ScheduleJobID

How to create

```
CREATE TABLE ScheduleJob(
 ScheduleJobID INTEGER NOT NULL,
 Type INTEGER,
 Id INTEGER,
 Addr1 NVARCHAR(255),
 Addr2 NVARCHAR(255),
 Occur INTEGER,
 Days NVARCHAR(255),
 ScheduleHour INTEGER,
 ScheduleMin INTEGER,
 ScheduleTime INTEGER,
 Cmd NVARCHAR(255),
 sParam NVARCHAR(max),
 iParam INTEGER,
 ScheduleOption INTEGER,
 TimeCreated INTEGER NOT NULL,
 TimeModified INTEGER NOT NULL,
 TimeDone INTEGER NOT NULL,
 Status INTEGER,
 Result INTEGER,
 SleepAfterSend INTEGER,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 MessageTitle NVARCHAR(255),
 RootOU INTEGER,
```

```
FriendlyName NVARCHAR(255),
PRIMARY KEY (ScheduleJobID));
```

6.3. Table: UpdateJob

This table holds the update jobs which are scheduled for execution or which have already been executed. If the **Status** value is **UPD_JOBSTATUS_RUNNING**, the entry may not be removed. To completely remove the update job, all referencing entries of table UpdateDevice must also be deleted.

ColumnName	DataType	Key	Not Null	Default Value	Comment
UpdateJobID	INTEGER	PK	NN		Unique ID of update job
Type	INTEGER				Job type **1** UPD_JOBTYPE_DEVICE — ID refers to Device **2** UPD_JOBTYPE_GROUP — ID refers to Groups
Id	INTEGER	(FK)			ID of update entity Depends on Type Refers to table Groups or Device
TimeCreated	INTEGER		NN		Job creation time (Unix time)
TimeModified	INTEGER		NN		Last modification (Unix time)
TimeUpdate	INTEGER		NN		Start time of update job (Unix time)
Status	INTEGER				Job state **0** UPD_JOBSTATUS_QUEUED — Job queued **1** UPD_JOBSTATUS_RUNNING — Job running **2** UPD_JOBSTATUS_DONE — Job done

ColumnName	DataType	Key	Not Null	Default Value	Comment
Result	INTEGER				Job result 0 UPD_JOBRESULT_OK Job result undefined 1 UPD_JOBRESULT_ERROR Job failed 2 UPD_JOBRESULT_CLOSED Job done
Command	NVARCHAR(max)				Last executed pseudo command (#UPDATE, #CUSTOMCOMMAND, #FORMAT)
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				
Administrator	NVARCHAR(255)				Admin who entered the command
FriendlyName	NVARCHAR(255)				Friendly name for command
RootOU	INTEGER				Required to check if the command history entries are within the admin's OU hierarchy

IndexName	IndexType	Columns
PRIMARY	PRIMARY	UpdateJobID

How to create

```
CREATE TABLE UpdateJob(
 UpdateJobID INTEGER NOT NULL,
 Type INTEGER,
 Id INTEGER,
 TimeCreated INTEGER NOT NULL,
 TimeModified INTEGER NOT NULL,
 TimeUpdate INTEGER NOT NULL,
 Status INTEGER,
 Result INTEGER,
 Command NVARCHAR(max),
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 Administrator NVARCHAR(255),
 FriendlyName NVARCHAR(255),
 RootOU INTEGER,
 PRIMARY KEY (UpdateJobID));
```

Stored Procedure

to retrieve values for the command history dialog

```
CREATE PROCEDURE UCSP_CommandHistory
@jobid as integer
AS
```

```
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON
-- Insert statements for procedure here
SELECT COUNT(CASE WHEN UpdateDevice.Status=5 AND UpdateDevice.Result=262 THEN 1 END) AS CountTimeout,
COUNT(CASE WHEN UpdateDevice.Status=5 THEN 1 END) AS CountDone,
COUNT(CASE WHEN UpdateDevice.Status=5 AND UpdateDevice.Result<>0 AND UpdateDevice.Result<>262 AND
UpdateDevice.Result<>2 THEN 1 END) AS CountFailed,
COUNT(CASE WHEN UpdateDevice.Status=5 AND (UpdateDevice.Result=0 OR UpdateDevice.Result=2) THEN 1 END) AS
CountSuccess, UpdateJob.UpdateJobID, UpdateJob.Type, Device.Name AS DeviceName,Groups.Name AS
GroupName,DynamicClientGroup.Name AS DCGName, UpdateJob.TimeCreated,UpdateJob.TimeModified, UpdateJob.Res-
ult,UpdateJob.Status,UpdateJob.Id,CAST(UpdateJob.Command as nvarchar(255)) as Command, UpdateJob.Ad-
ministrator as Administrator,
COUNT(UpdateDevice.UpdateJobID) as DeviceCount,
ConfigRun.DeviceCount AS ConfigRunDeviceCount,
ConfigRun.DevicesDone AS ConfigRunDevicesDone
FROM (UpdateJob
LEFT JOIN Groups ON (UpdateJob.Id=Groups.GroupID))
LEFT JOIN Device ON (UpdateJob.Id=Device.DeviceID)
LEFT JOIN UpdateDevice ON (UpdateDevice.UpdateJobID=UpdateJob.UpdateJobID)
LEFT JOIN DynamicClientGroup ON (UpdateJob.Id = DynamicClientGroup.DynamicClientGroupID)
LEFT JOIN ConfigRun ON (UpdateJob.UpdateJobId = ConfigRun.UpdateJobID)
WHERE (UpdateJob.UpdateJobID = @jobid AND @jobid > -1) OR (UpdateJob.UpdateJobID > -1 and @jobid=-1)
GROUP BY UpdateJob.UpdateJobID, UpdateJob.Type, Device.Name, Groups.Name, DynamicClientGroup.Name,
UpdateJob.TimeCreated, UpdateJob.TimeModified, UpdateJob.Result, UpdateJob.Status, UpdateJob.Id, CAST
(UpdateJob.Command as nvarchar(255)), UpdateJob.Administrator, ConfigRun.DeviceCount, Con-
figRun.DevicesDone
END
```

```
;
```

6.4. Table: UpdateDevice

Each entry holds the data of a single device which is (or was) part of an update job. Entries may only be removed if the corresponding update job is not running.

ColumnName	Data Type	Key	Not Null	Default Value	Comment
UpdateDeviceID	INTEGER	PK	NN		Unique ID of update device
UpdateJobID	INTEGER	FK			The Id of update job this client is part of Foreign key refers to UpdateJob table
Id	INTEGER	(FK)	NN		Client ID Refers to DeviceID of **Device** table
Mac	NVARCHAR(255)		NN		Client MAC address
ClientIdentifier	UNIQUEIDENTIFIER		NN		Unique identifier for client devices and eLux portable sticks
Ip	NVARCHAR(255)		NN		Client IP address
TimeCreated	INTEGER		NN		Creation time (Unix time)
TimeModified	INTEGER		NN		Last modification time (Unix time)
SetupID	INTEGER	(FK)			Device configuration where the firmware parameters come from (refers to table Setup)

ColumnName	Data Type	Key	Not Null	Default Value	Comment
UpdOption	INTEGER			0	Update option **1** UPD_DEVOPTION_V2 **2** UPD_DEVOPTION_V6 **4** UPD_DEVOPTION_CANFORMAT **7** Combination of the flags above
StartStatus	INTEGER				State of device at beginning of update job **1** UPD_DEVSTATUS_OFF Client was switched off **3** UPD_DEVSTATUS_ON Client was up and running
Status	INTEGER				Current state of device **1** UPD_DEVSTATUS_OFF Client is switched off **2** UPD_DEVSTATUS_STARTING Client is starting **3** UPD_DEVSTATUS_ON Client is up and running **4** UPD_DEVSTATUS_UPDATING Client is updating **5** UPD_DEVSTATUS_DONE Client has been updated **6** UPD_DEVSTATUS_WAITHALT Client is shutting down

ColumnName	Data Type	Key	Not Null	Default Value	Comment
Result	INTEGER				Update result **0** UPD_JOBRESULT_OK — Update successful **1** UPD_JOBRESULT_ERROR — Update failed
GroupID	INTEGER	(FK)			Group which owns the device specific update parameters (foreign key refers to table Groups)
Version	NVARCHAR(255)				Version of client
ProgressStatus	INTEGER				Status of update process: Start transfer, End transfer, Deferred, Start installation, End installation
ClientFeatures	INTEGER				Features the client is capable of (can be a bit combination of values) For the values, see the Device table.
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	UpdateDeviceID

How to create

```
CREATE TABLE UpdateDevice(
UpdateDeviceID INTEGER NOT NULL,
UpdateJobID INTEGER,
Id INTEGER NOT NULL,
Mac NVARCHAR(255) NOT NULL,
ClientIdentifier UNIQUEIDENTIFIER NOT NULL,
Ip NVARCHAR(255) NOT NULL,
TimeCreated INTEGER NOT NULL,
TimeModified INTEGER NOT NULL,
SetupID INTEGER,
UpdOption INTEGER,
StartStatus INTEGER,
Status INTEGER,
Result INTEGER,
GroupID INTEGER,
Version NVARCHAR(255),
ProgressStatus INTEGER,
ClientFeatures INTEGER;
ReservedString1 NVARCHAR(255),
ReservedString2 NVARCHAR(255),
ReservedInt1 INTEGER,
ReservedInt2 INTEGER,
CONSTRAINT FK_UpdateDevice_UpdateJobID_UpdateJob_UpdateJobID FOREIGN KEY (UpdateJobID) REFERENCES
UpdateJob (UpdateJobID),
CONSTRAINT PK_UpdateDevice
PRIMARY KEY (UpdateDeviceID));
```

6.5. Table: UpdateResult

Each entry represents an firmware update result (including complete update log file) according to an device specified by MAC address.

ColumnName	DataType	Key	Not Null	Default Value	Comment
UpdateResultID	INTEGER	PK	NN		Unique ID of update result
TimeReceived	INTEGER		NN		Entry timestamp (Unix time)
Mac	NVARCHAR(12)		NN		Client MAC address
ClientIdentifier	UNIQUEIDENTIFIER		NN		Unique identifier for client devices and eLux portable sticks

ColumnName	DataType	Key	Not Null	Default Value	Comment
IntResult	INTEGER				Result of last update **0** UPD_DEVRESULT_OK Update successful **1** ~_E_UNKNOWN Update failed due to unspecified reason **2** ~_UPTODATE Update not necessary, client up to date **3** ~_INPROGRESS Update still in progress **256** ~_E_GENERAL Update failed, error unknown **257** ~_E_FIRMPARAM Update failed, bad firmware parameter **258** ~_E_AUTH Update failed, authorization failed **259** ~_E_FLASHSIZE Update failed, unsufficient flash size **260** ~_E_CONTAINER Update failed, container mismatch **261** ~_E_UNREACHED Update failed, client unreachable **262** ~_E_TIMEOUT Update failed, no answer from client
TextResult	NVARCHAR(max)				Update log file received from client
CommandType	INTEGER			0	0: Update 1: Custom command
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	UpdateResultID

How to create

```
CREATE TABLE UpdateResult(
 UpdateResultID INTEGER NOT NULL,
 TimeReceived INTEGER NOT NULL,
 Mac NVARCHAR(12) NOT NULL,
 ClientIdentifier UNIQUEIDENTIFIER NOT NULL,
 IntResult INTEGER,
 TextResult NVARCHAR(max),
 CommandType INTEGER,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 CONSTRAINT PK_UpdateResult
 PRIMARY KEY (UpdateResultID));
```

7. Scout Enterprise Intern

7.1. Table: Administrator

This table contains two kinds of entries:

1. Scout Enterprise administrator profiles (identified by setting Type=-1 and ID=-1)
2. Administrator policies/object rights on either a device or an OU/group (identified by Type=1|4 and ID>=0)

The **Rights** attribute is a 210 bit long string, each bit represents a single permission. In order to create Administrator entries manually, it is good practice to add a new entry via Scout Enterprise and copy the **Rights** value to your new entry.

ColumnName	DataType	Key	Not Null	Default Value	Comment
AdministratorID	INTEGER	PK	NN		Unique ID of administrator entry
Name	NVARCHAR (255)				Name of real user or group (f.e. <server>\<user>)
Sid	NVARCHAR (255)				- not used -
Rights	NVARCHAR (255)			0000000000000000000000000000000000000000000000000000	Definition of policy/rights

ColumnName	DataType	Key	Not Null	Default Value	Comment
Type	INTEGER				**-1** Administrator profile **1** TYPE_GROUP Permissions on a Scout Enterprise OU (ID refers to Groups) **4** TYPE_DEVICE_ENTRY Permissions on a Scout Enterprise device (ID refers to Device)
ID	INTEGER				**-1** Administrator profile **>=0** Scout Enterprise object ID For Type=1, the foreign key refers to table Device.DeviceID, for Type=4, the foreign key refers to table Groups.GroupID
ProfileType	SMALLINT(6)				Pre-defined profiles (administrator profiles only): **0** Minimum profile (objects not visible, menus not allowed) **1** Scout Enterprise object ID **3** Individual profile
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				

ColumnName	DataType	Key	Not Null	Default Value	Comment
ReservedInt1	INTEGER			0	Optional ID of root OU Refers to table Groups.GroupID
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	AdministratorID

HOW TO CREATE

```
CREATE TABLE Administrator(
 AdministratorID INTEGER NOT NULL,
 Name NVARCHAR(255),
 Sid NVARCHAR(255),
 Rights NVARCHAR(255),
 Type INTEGER,
 ID INTEGER,
 ProfileType SMALLINT,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (AdministratorID));
```

7.2. Table: Alert

Scout Enterprise alert definitions. For internal use only.

ColumnName	DataType	Key	Not Null	Default Value	Comment
AlertID	INTEGER IDENTITY (1,1)	PK	NN		Unique ID of alert entry
EntityType	INTEGER				
IPAddress	NVARCHAR(255)				
IPName	NVARCHAR(255)				
AlertIndex	INTEGER				
AlertText	NVARCHAR(max)				
AlertTime	INTEGER				
AlertStatus	INTEGER				
AlertType	INTEGER				
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	AlertID

HOW TO CREATE

```
CREATE TABLE Alert(
 AlertID INTEGER NOT NULL,
 EntityType INTEGER,
 IPAddress NVARCHAR(255),
 IPName NVARCHAR(255),
 AlertIndex INTEGER,
 AlertText NVARCHAR(max),
 AlertTime INTEGER,
 AlertStatus INTEGER,
 AlertType INTEGER,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (AlertID));
```

7.3. Table: ConsoleAction

Console actions are a integral part of the internal communications between Scout Enterprise components. Table entries should never be manually added, removed or changed.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ConsoleActionID	INTEGER IDENTITY (1,1)	PK	NN		Unique ID of console action entry
Type	INTEGER		NN		
ConsoleID	INTEGER	(FK)	NN		Target console of the action (refers to EntityID of table Entity)
MemberType	INTEGER				
MemberID	INTEGER				
StringParameter	NVARCHAR(255)				
IntParameter	INTEGER				
StringParameter2	NVARCHAR(255)				
IntParameter2	INTEGER				
StringParameter3	NVARCHAR(255)				
IntParameter3	INTEGER				
StringParameter4	NVARCHAR(255)				
IntParameter4	INTEGER				
StringParameter5	NVARCHAR(255)				
IntParameter5	INTEGER				
IntParameter6	INTEGER				
IntParameter7	INTEGER				
IntParameter8	INTEGER				
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ConsoleActionID

HOW TO CREATE

```
CREATE TABLE ConsoleAction(
 ConsoleActionID INTEGER NOT NULL,
 Type INTEGER NOT NULL,
 ConsoleID INTEGER,
 MemberType INTEGER,
 MemberID INTEGER,
 StringParameter NVARCHAR(255),
 IntParameter INTEGER,
 StringParameter2 NVARCHAR(255),
 IntParameter2 INTEGER,
 StringParameter3 NVARCHAR(255),
 IntParameter3 INTEGER,
 StringParameter4 NVARCHAR(255),
 IntParameter4 INTEGER,
 StringParameter5 NVARCHAR(255),
 IntParameter5 INTEGER,
 IntParameter6 INTEGER,
 IntParameter7 INTEGER,
 IntParameter8 INTEGER,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (ConsoleActionID));
```

7.4. Table: DynamicClientGroup

ColumnName	DataType	Key	Not Null	Default Value	Comment
DynamicClientGroupID	INTEGER	PK	NN		Unique ID of Dynamic Client Group entry
Name	NVARCHAR (255)				Name of the Dynamic Client Group
Query	NVARCHAR (max)				Last filter phrase of the SQL query
CreationDate	INTEGER				Creation date of the Dynamic Client Group
ScoutAdmin	NVARCHAR (255)				Scout Enterprise administrator who created the Dynamic Client Group
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	DynamicClientGroupID

HOW TO CREATE

```
CREATE TABLE DynamicClientGroup(
 DynamicClientGroupID INTEGER NOT NULL,
 Name NVARCHAR(255),
 Query NVARCHAR(max),
 CreationDate INTEGER,
 ScoutAdmin NVARCHAR(255),
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (DynamicClientGroupID));
```

7.5. Table: DynamicClientGroupDevices

ColumnName	DataType	Key	Not Null	Default Value	Comment
DynamicClientGroupDevicesID	INTEGER	PK	NN		Unique ID of Dynamic Client Group devices
DynamicClientGroupID	INTEGER				ID of the Dynamic Client Group the device belongs to
Mac	NVARCHAR(255)				MAC address of the device
ClientIdentifier	UNIQUEIDENTIFIER		NN		Unique identifier for client devices and eLux portable sticks
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	DynamicClientGroupDevicesID

HOW TO CREATE

```
CREATE TABLE DynamicClientGroupDevices(
 DynamicClientGroupDevicesID INTEGER NOT NULL,
 DynamicClientGroupID INTEGER,
 Mac NVARCHAR(255),
 ClientIdentifier UNIQUEIDENTIFIER,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (DynamicClientGroupDevicesID));
```

7.6. Table: Entity

The Entries in this table correspond to running Scout Enterprise components like Scout Enterprise server and console. Since the entries are created and removed dynamically, it is not recommended to change them.

ColumnName	DataType	Key	Not Null	DefaultValue	Comment
EntityID	INTEGER	PK	NN		Unique ID of entity entry
Type	INTEGER				Entity type **0** Scout Enterprise Server **1** Scout Enterprise Console
Version	NVARCHAR (255)				Entity version
IPAddress	NVARCHAR (255)				
StartTime	INTEGER				
HostID	NVARCHAR (64)				
Username	NVARCHAR (255)				
Features	INTEGER				
Mode	INTEGER				
HeartBeat	INTEGER				
Status	INTEGER				
Threshold1	INTEGER				
Threshold2	INTEGER			90	
Threshold3	INTEGER				
Threshold4	INTEGER				
DatabaseConnections	INTEGER				

ColumnName	DataType	Key	Not Null	DefaultValue	Comment
IPName	NVARCHAR (255)				
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				
IccInstanceID	INTEGER				
LogFileName	NVARCHAR (255)				Name of log file used by an entity

IndexName	IndexType	Columns
PRIMARY	PRIMARY	EntityID

HOW TO CREATE

```
CREATE TABLE Entity(
 EntityID INTEGER NOT NULL,
 Type INTEGER NOT NULL,
 Version NVARCHAR(255),
 IPAddress NVARCHAR(255),
 StartTime INTEGER,
 HostID NVARCHAR(64),
 Username NVARCHAR(255),
 Features INTEGER,
 Mode INTEGER,
 HeartBeat INTEGER,
 Status INTEGER,
 Threshold1 INTEGER,
 Threshold2 INTEGER,
 Threshold3 INTEGER,
 Threshold4 INTEGER,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 IccInstanceID INTEGER,
 IPName NVARCHAR(255),
 DatabaseConnections INTEGER,
 LogFileName NVARCHAR(255),
 PRIMARY KEY (EntityID));
```

7.7. Table: IccCommand

This table store all commands for the inter console communication

ColumnName	DataType	Key	Not Null	Default Value	Comment
IccCommandID	INTEGER	PK	NN		
IdCmdID	INTEGER				
CreatorInstanceID	INTEGER				
CreatedAt	INTEGER				
InstanceID	INTEGER				
Command	INTEGER				
CommandParamID	INTEGER				
ValidFrom	INTEGER				
ValidTo	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	IccCommandID

HOW TO CREATE

```
CREATE TABLE IccCommand(
IccCommandID INTEGER NOT NULL,
IsCmdID INTEGER,
CreatorInstanceID INTEGER,
CreatedAt INTEGER,
InstanceID INTEGER,
Command INTEGER,
CommandParamID INTEGER,
ValidFrom INTEGER,
ValidTo INTEGER,
DoneAt INTEGER,
PRIMARY KEY (IccCommandID));
```

7.8. Table: IccMessageParam

This table contains all messages which may be sent from one to another console

ColumnName	DataType	Key	Not Null	Default Value	Comment
IccMessageParam	INTEGER	PK	NN		
DisplayTime	INTEGER				
CanCancel	SMALLINT				
MsgText1	NVARCHAR(255)				
MsgText2	NVARCHAR(255)				
MsgText3	NVARCHAR(255)				
MsgText4	NVARCHAR(255)				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	IccMessageParam

HOW TO CREATE

```
CREATE TABLE IccMessageParam(
IccMessageParamID INTEGER NOT NULL,
DisplayTime INTEGER,
CanCancel SMALLINT,
MsgText1 NVARCHAR(255),
MsgText2 NVARCHAR(255),
MsgText3 NVARCHAR(255),
MsgText4 NVARCHAR(255),
PRIMARY KEY (IccMessageParamID));
```

7.9. Table: IccTerminateParam

This table store all terminate requests from one console to another

ColumnName	DataType	Key	Not Null	Default Value	Comment
IccTerminateParam	INTEGER	PK	NN		
DisplayTime	INTEGER				
CanCancel	SMALLINT				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	IccTerminateParam

HOW TO CREATE

```
CREATE TABLE IccTerminateParam(
IccTerminateParamID INTEGER NOT NULL,
DisplayTime INTEGER,
CanCancel SMALLINT,
PRIMARY KEY (IccTerminateParamID));
```

7.10. Table: LicenseMonitor

Table for new license log in Scout Enterprise Console 15

ColumnName	DataType	Key	Not Null	Default Value	Comment
LicenseMonitorID	INTEGER IDENTITY (1,1)	PK	NN		Unique ID
MonitorUser	NVARCHAR (255)				
MonitorTime	DATETIME		NN	GETUTCDATE ()	
Category	INTEGER		NN	0	0: no category 1: asset - license monitor entry for an action changing the license asset (e.g adding a license) 2: assignment - license monitor entry for an action changing the assignment of licenses 4: message - license monitor entry is an information message
Reason	INTEGER		NN	0	Examples: ADD_DEVICE_ MANUAL, ADD_DEVICE_AUTO, RELOCATE_LEAVE, RELOCATE_ENTER...
Product	INTEGER		NN	0	Product ID the license monitor entry is related to
Comment	NVARCHAR (255)				
CompleteCount	INTEGER		NN	0	Total of licenses of this type
DiffCount	INTEGER		NN	0	Difference in number
UsedCount	INTEGER		NN	0	Number of used licenses
ValidDays	INTEGER		NN	0	Validity period in days for temporary licenses
HostID	NVARCHAR (255)		NN	' '	Host ID of added licenses

ColumnName	DataType	Key	Not Null	Default Value	Comment
ActivationKey	NVARCHAR (255)			''	Activation key of added licenses
LicenseKey	NVARCHAR (255)		NN	''	License key of added licenses

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ScoutLicenseID

HOW TO CREATE

```
CREATE TABLE LicenseMonitor (
 LicenseMonitorID INTEGER IDENTITY (1,1) NOT NULL,
 MonitorUser NVARCHAR(255)
 MonitorTime DATETIME NOT NULL DEFAULT(GETUTCDATE()),
 Category INTEGER NOT NULL DEFAULT 0,
 Reason INTEGER NOT NULL DEFAULT 0,
 Product INTEGER NOT NULL DEFAULT 0,
 Comment NVARCHAR(255),
 CompleteCount INTEGER NOT NULL DEFAULT 0,
 DiffCount INTEGER NOT NULL DEFAULT 0,
 UsedCount INTEGER NOT NULL DEFAULT 0,
 ValidDays INTEGER NOT NULL DEFAULT 0,
 HostID VARCHAR(255) NOT NULL DEFAULT '',
 ActivationKey VARCHAR(255) NULL DEFAULT '',
 LicenseKey VARCHAR(255) NOT NULL DEFAULT '',
 CONSTRAINT PK_LicenseMonitor,
 PRIMARY KEY (LicenseMonitorID));
```

7.11. Table: LockList

In this table database and table locks are managed by Scout Enterprise Server and Console. Do not make any modifications.

ColumnName	DataType	Key	Not Null	Default Value	Comment
LockListID	INTEGER IDENTITY(1,1)	PK	NN		Unique ID of lock list entry Autoincrement value for Scout Enterprise 14.5.0 and later versions
Type	INTEGER				
Id	INTEGER				
Mode	INTEGER				
ConsoleID	INTEGER				
ServerID	INTEGER				
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	LockListID

```
HOW TO CREATE

CREATE TABLE LockList(
 LockListID INTEGER IDENTITY(1,1) NOT NULL,
 Type INTEGER,
 Id INTEGER,
 Mode INTEGER,
 ConsoleID INTEGER,
 ServerID INTEGER,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (LockListID));
```

7.12.

7.13. Table: MaintenanceWindow

Holds the data for command preferences (for standard commands).

ColumnName	DataType	Key	Not Null	Default Value	Comment
MaintenanceWindowID	INTEGER IDENTITY(1,1)	PK	NN		Unique maintenance window ID
From	DATETIME		NN		Start time of maintenance window
To	DATETIME		NN		End time of maintenance window
Repeat	BIT		NN		Is maintenance window repeating?

ColumnName	DataType	Key	Not Null	Default Value	Comment
RepeatType	INTEGER		NN		Repeating period: 0 REPEAT_DAILY 1 REPEAT_WEEKLY 2 REPEAT_MONTHLY 3 REPEAT_MONTHLY_FIRST_WEEKDAY 4 REPEAT_MONTHLY_SECOND_WEEKDAY 5 REPEAT_MONTHLY_THIRD_WEEKDAY 6 REPEAT_MONTHLY_FOURTH_WEEKDAY 7 REPEAT_MONTHLY_LAST_WEEKDAY
Name	NVARCHAR (255)				Name of maintenance window
Administrator	NVARCHAR (255)				Name of administrator who has defined the maintenance window
Modified	DATETIME		NN		Last modification of maintenance window

IndexName	IndexType	Columns
PRIMARY	PRIMARY	MaintenanceWindowID

HOW TO CREATE

```
CREATE TABLE MaintenanceWindow(
MaintenanceWindowID __UNICON_AUTOINCREMENT__ NOT NULL,
[From] DATETIME not null,
[To] DATETIME not null,
[Repeat] BIT not null,
RepeatType INTEGER not null,
[Name] NVARCHAR(255),
Administrator NVARCHAR(255),
Modified DATETIME not null,
CONSTRAINT PK_MaintenanceWindow
PRIMARY KEY (MaintenanceWindowID));
```

7.14. Table: Monitor

ColumnName	DataType	Key	Not Null	Default Value	Comment
MonitorID	INTEGER	PK	NN		Unique ID of monitor entry
MonitorType	INTEGER				
MonitorUser	NVARCHAR(255)				
ObjectName	NVARCHAR(max)				
MonitorTime	INTEGER				
Reason	INTEGER				
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	MonitorID

HOW TO CREATE

```
CREATE TABLE Monitor(
 MonitorID INTEGER NOT NULL,
 MonitorType INTEGER,
 MonitorUser NVARCHAR(255),
 ObjectName NVARCHAR(max),
 MonitorTime INTEGER,
 Reason INTEGER,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (MonitorID));
```

7.15. Table: OUByNet

Entries in this table correspond to running Scout Enterprise components like Scout Enterprise server and console. Since the entries are created and removed dynamically, it is not recommended to change them.

ColumnName	DataType	Key	Not Null	Default Value	Comment
OUByNetID	INTEGER	PK	NN		Unique ID
FilterType	INTEGER				Type of filter. 1: Subnet filter 2: Custom filter
SubnetAddr	NVARCHAR (255)				Subnet address if FilterType = 1
CustomFilter	NVARCHAR (max)				Custom filter string if FilterType i= 2
OUID	INTEGER				ID of the OU to which devices within the subnet address should be assigned
OrderNumber	INTEGER				Order number for processing the rules
ReservedString1	NVARCHAR (255)				
ReservedString2	NVARCHAR (255)				
INTEGER	INTEGER				
ReservedInt2	INTEGER				
Active	BIT		NN	1	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	OUByNetID

How to create

```
CREATE TABLE OUByNet(
 OUByNetID INTEGER NOT NULL,
 FilterType INTEGER,
 SubnetAddr NVARCHAR(255),
 CustomFilter NVARCHAR(max),
 OUID INTEGER,
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
Active BIT NOT NULL,
PRIMARY KEY (OUByNetID));
```

7.16. Table: OUsApps

Assignment table holding the information about which application is inherited to the OU below.

ColumnName	DataType	Key	Not Null	Default Value	Comment
OUSAppsID	INTEGER IDENTITY (1,1)	PK	NN		
OUID	INTEGER		NN		
AppID	INTEGER		NN		
OUOrigin	INTEGER		NN		

IndexName	IndexType	Columns
PRIMARY	PRIMARY	OUSAppsID

HOW TO CREATE

```
CREATE TABLE OUSApps(
OUSAppsID INTEGER IDENTITY (1,1) NOT NULL,
OUID INTEGER NOT NULL,
AppID INTEGER NOT NULL,
OUOrigin INTEGER NOT NULL,
CONSTRAINT PK_OUSApps
PRIMARY KEY (OUSAppsID));
```

7.17. Table: PredefinedCommandTemplates

Data for command preferences (for standard commands)

ColumnName	DataType	Key	Not Null	Default Value	Comment
CommandID	INTEGER	PK	NN		Unique ID of the command **1** Shutdown **3** Restart **5** Update **8** Factory reset **10** Delivery

ColumnName	DataType	Key	Not Null	Default Value	Comment
Policies	INTEGER		NN	0	Values for preferences, can be combined by OR: **0x8** Inform user **0x100** Format before update **0x200** Run as root **0x800** Enable this template **0x1000** Template is mandatory **0x2000** User can cancel
InformInSeconds	INTEGER		NN	0	Number of seconds to show the command message

IndexName	IndexType	Columns
PRIMARY	PRIMARY	CommandID

HOW TO CREATE

```
CREATE TABLE PredefinedCommandTemplates(
CommandID INTEGER NOT NULL,
Policies INTEGER NOT NULL DEFAULT 0,
InformInSeconds INTEGER NOT NULL DEFAULT 0,
CONSTRAINT PK_PredefinedCommandTemplates
PRIMARY KEY (CommandID));
```

HOW TO INITIALIZE

```
INSERT INTO PredefinedCommandTemplates (CommandID) VALUES (1);
INSERT INTO PredefinedCommandTemplates (CommandID) VALUES (3);
INSERT INTO PredefinedCommandTemplates (CommandID) VALUES (5);
INSERT INTO PredefinedCommandTemplates (CommandID) VALUES (8);
INSERT INTO PredefinedCommandTemplates (CommandID) VALUES (10);
```

7.18. Table: ReportsAdmins

Assignment table holding the information about which administrator is allowed to execute which report.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ReportsAdminsID	INTEGER IDENTITY (1,1)	PK	NN		
AdminID	INTEGER		NN		

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ReportsAdminsID

HOW TO CREATE

```
CREATE TABLE ReportsAdmins(
ReportsAdminsID INTEGER IDENTITY (1,1) NOT NULL,
AdminID INTEGER NOT NULL,
CONSTRAINT PK_ReportsAdmins
PRIMARY KEY (ReportsAdminsID));
```

7.19. Table: RptDefApp

This table contains all database fields of an application which should be visible to the Scout Enterprise report tool.

ColumnName	DataType	Key	Not Null	Default Value	Comment
Field	NVARCHAR(255)	PK	NN		
Access	NVARCHAR(255)				
Text_DE	NVARCHAR(255)				
Text_EN	NVARCHAR(255)				
Type	NVARCHAR(255)				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	Field

HOW TO CREATE

```
CREATE TABLE RptDefApp(
Field NVARCHAR(255) NOT NULL,
Access NVARCHAR(255) NULL,
Text_DE NVARCHAR(255) NULL,
Text_EN NVARCHAR(255) NULL,
Type NVARCHAR(255) NULL,
PRIMARY KEY (Field));
```

HOW TO INITIALIZE

```
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('AppOnDesktop', null, 'Desktop Symbol', 'Desktop icon',
'b');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('AutoStart', null, 'Autostart', 'Autostart', 'b');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_AppType', 'CALC_AppTypes.AppTypeID=AppTypeID,AppType',
'Typ', 'Type', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_Browser_CallPage', 'Browser-
.ApplicationID=ApplicationID,CallPage', 'Browser Seite', 'Browser
```

```
Page', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_FQName', 'CALC_FQName', 'Name mit Pfad', 'Name and
path', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_Groupname', 'Groups.GroupID=GroupID,Name', 'Organ-
isationseinheit', 'Organisation unit', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_ICA_Application', 'ICA.Ap-
plicationID=ApplicationID,Application', 'ICA Anwendung', 'ICA
Application', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_ICA_Server', 'ICA.ApplicationID=ApplicationID,Server',
'ICA Server', 'ICA Server', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_PNA_Server', 'PNA.Ap-
plicationID=ApplicationID,ServerUrl', 'PNA Server', 'PNA Server',
's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_RDP_Application', 'RDP.Ap-
plicationID=ApplicationID,Application', 'RDP Anwendung', 'RDP
Application', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_RDP_Server', 'RDP.ApplicationID=ApplicationID,Server',
'RDP Server', 'RDP Server', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_VD_Server', 'VDA.ApplicationID=ApplicationID,Server',
'VD Server', 'VD Server', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Name', null, null, null, 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('ReservedInt1', null, 'Autostart nach', 'Autostart delay',
'i');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Roaming', null, null, null, 'b');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('TM', null, 'Dauerbetrieb', 'Restart', 'b');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('DisplayName', null, 'Anzeigename', 'Display name', 's');
```

7.20. Table: RptDefAsset

This table contains all database fields of an asset which should be visible to the Scout Enterprise report tool.

ColumnName	DataType	Key	Not Null	Default Value	Comment
Field	NVARCHAR(255)	PK	NN		
Access	NVARCHAR(255)				
Text_DE	NVARCHAR(255)				
Text_EN	NVARCHAR(255)				
Type	NVARCHAR(255)				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	Field

HOW TO CREATE

```
HOW TO CREATE TABLE RptDefAsset(
Field NVARCHAR(255) NOT NULL,
Access NVARCHAR(255) NULL,
Text_DE NVARCHAR(255) NULL,
Text_EN NVARCHAR(255) NULL,
Type NVARCHAR(255) NULL,
PRIMARY KEY (Field));
```

HOW TO INITIALIZE

```
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('AssetName', null, 'Name', 'Name', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_AssetValue', 'CALC_AssetValue', 'Typ', 'Type', 'i');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('DeviceID', null, 'Geräte ID', 'Device ID', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_ProductID', 'CALC_ProductID', 'Produkt ID', 'Product
ID', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_VendorID', 'CALC_VendorID', 'Hersteller ID', 'Vendor
```

```
ID', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_ProductText', 'CALC_ProductText', 'Beschreibung',
'Description', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_VendorText', 'CALC_VendorText', 'Hersteller', 'Vendor',
 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_Serial', 'CALC_Serial', 'Seriennummer', 'Serial num-
ber', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_DeviceName', 'Device.DeviceID=DeviceID,Name', 'Ger-
ätename', 'Device name', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_DeviceMAC', 'Device.DeviceID=DeviceID,Mac_Address',
'MAC Adresse', 'MAC address', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_FQName', 'CALC_FQName', 'Gerätename mit Pfad', 'Device
name with path', 's');
```

7.21. Table: RptDefDevice

This table contains all database fields of a device which should be visible to the Scout Enterprise report tool.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ColumnName	DataType	Key	NN	Default Value	Comment
Field	NVARCHAR(255)	PK	NN		
Access	NVARCHAR(255)				
Text_DE	NVARCHAR(255)				
Text_EN	NVARCHAR(255)				
Type	NVARCHAR(255)				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	Field

How to create

```
CREATE TABLE RptDefDevice(
Field NVARCHAR(255) NOT NULL,
Access NVARCHAR(255) NULL,
Text_DE NVARCHAR(255) NULL,
Text_EN NVARCHAR(255) NULL,
Type NVARCHAR(255) NULL,
PRIMARY KEY (Field));
```

How to initialize

```
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Activated', 'CALC_TransferLockActivated', 'Kon-
figurationstransfer', 'Configuration transfer', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('ActiveUser', null, 'Benutzer', 'User', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('BiosVersion', null, 'Bios', 'Bios', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Container', null, null, null, 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
```

```
VALUES ('EX_CpuMHz', 'CALC_CpuMHz', 'CPU (MHz)', 'CPU (MHz)', 'i');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('CreationDate', null, 'Erstellungsdatum', 'Creation date',
'u');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('DeviceType', null, 'Gerätetyp', 'Device type', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Domain', null, null, null, 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_FQName', 'CALC_FQName', 'Name mit Pfad', 'Name and
path', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_Groupname', 'Groups.GroupID=GroupID,Name', 'Organ-
isationseinheit', 'Organisation unit', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_Keyboard', 'XkbLayout.XkbLayoutID=CALC_XkbLay-
outID,Name', 'Tastatur', 'Keyboard', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_NumLocalPrinters', 'CALC_NumLocalPrinters', 'Anzahl
lokale Drucker', 'Number of local printers', 'i');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_NumNetPrinters', 'CALC_NumNetPrinters', 'Anzahl Net-
zwerkdrucker', 'Number of network printers', 'i');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_NumPrinters', 'CALC_NumPrinters', 'Anzahl Drucker',
'Number of printers', 'i');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_Printer', 'CALC_DefaultPrinter', 'Standarddrucker',
'Default printer', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_USBEnabled', 'CALC_UseUSB', 'USB zugelassen', 'USB
enabled', 'b');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EluxManager', null, 'Manager', 'Manager', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('FlashSize', null, 'Flashgröße', 'Flashsize', 'i');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('FlashType', null, 'Flashspeicher', 'Flash memory', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('IP_Address', null, 'IP-Adresse', 'IP address', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('ImageFile', 'Dis-
tri-
butionImageFile.Dis-
tributionImageFileID=DistributionImageFileID,ImageFile', 'Image',
'Image', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
```

```
VALUES ('Info1', null, null, null, 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Info2', null, null, null, 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Info3', null, null, null, 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('LastContact', null, 'Letzter Kontakt', 'Last contact',
'u');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Mac_Address', null, 'MAC-Adresse', 'MAC address', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('MemorySize', null, 'Hauptspeicher', 'Main memory', 'i');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('MonitorInfo', null, 'Monitorinfo', 'Monitor Info', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('MotherBoard', null, 'Typ', 'Type', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Name', null, null, null, 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Netmask', null, 'Netzmaske', null, 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('OS', null, null, null, 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('OSVersion', null, 'OS-Version', 'OS version', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('ScreenInfo', null, 'Monitorstatus', 'Screen info', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('SerialNumber', null, 'Seriennummer', 'Serial number',
's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Supplier', null, 'Hersteller', 'Supplier', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('VersionString', null, 'Version', 'Version', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('NetworkSpeed', null, 'Netzwerkgeschwindigkeit', 'Network
speed', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('OUByNet', 'CALC_OUByNet', 'OU-Filter aktiv', 'OU filter
active', 'b');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('OUAssignment','CALC_OUAssignment','OU-Zuweisung', 'OU-
Assignment', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_PowertermLicense','CALC_License_56' , 'Anwendungslizenz
: Powerterm' ,'Application license : powerterm' ,'b');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
```

```
VALUES ('EX_X97License' ,'CALC_License_18' , 'Anwendungslizenz :
X97-Emulation' ,'Application license : X97 Emulation' ,'b');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_X3252License' ,'CALC_License_20' , 'Anwendungslizenz :
3270/X5250-Emulation','Application license : X3270/X5250 Emu-
lation','b');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_LastCommandName' ,'CALC_LastCommandName' , 'Kom-
mandoname','Command name','s');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_LastCommandResult' ,'CALC_LastCommandResult' , 'Kom-
mandergebnis','Command result','s');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_LastCommandTime' ,'CALC_LastCommandTime' , 'Kom-
mandzeit','Command time','u');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_ActiveMac' ,'CALC_ActiveMac' ,'Aktive MAC-Adresse',
'Active MAC address' , 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_Media' ,'CALC_Media' ,'Medium' ,'Media' ,'s');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_LastUpdateSize', 'CALC_LastUpdateSize', 'Updategröße
(kB)', 'Update size (kB)', 'i');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('OUCustomFilter', null, 'OU Filtertext', 'OU filter text',
's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_Relocation', 'CALC_Relocation', 'Umzugsvormerkung',
'Relocation notification', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_UpdatePartitionSize', 'CALC_UpdatePartitionSize',
'Updatepartition (MB)', 'Update partition (MB)', 'i');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_SystemPartitionSize', 'CALC_SystemPartitionSize', 'Sys-
tempartition (MB)', 'System partition (MB)', 'i');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_ConfigDataUpToDate','CALC_Con-
figDataUpToDate','Konfiguration aktuell','Configuration up to
date', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_LastDeliveryTime' ,'CALC_LastDeliveryTime' ,
'Auslieferzeitpunkt','Delivery time','u');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_LastDeliveryResult' ,'CALC_LastDeliveryResult' ,
'Auslieferergebnis','Delivery result','s');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
```

```
VALUES ('EX_LastDeliverySize' ,'CALC_LastDeliverySize' , 'Ausliefer-
größe (kB)','Delivery size (kB)','i');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('ConfigDataID' ,NULL, 'Konfigurations-ID','Configuration
id','s');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('ConfigDataIDToHave' ,NULL, 'Erwartete Konfigurations-
ID','Expected configuration id','s');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_LastUpdateProvider', 'CALC_LastUpdateProvider', 'Update-
provider', 'Update provider', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_LastDeliveryProvider', 'CALC_LastDeliveryProvider',
'Auslieferprovider', 'Delivery provider', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('DeliveryImage', null, 'Auslieferimage', 'Delivery image',
's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_UpdateNotification', 'CALC_UpdateNotification', 'Firm-
wareaktualisierungsvormerkung', 'Image update notification', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('NetAddr', null, 'Netzadresse', 'Net address', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_FluendoLicenseBasic','CALC_License_91',
'Anwendungslizenz : Fluendo','Application license : Fluendo', 'b');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_FluendoLicenseSystem','CALC_License_92',
'Anwendungslizenz : Fluendo systemweit','Application license :
Fluendo System-wide', 'b');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Note', null, 'Hinweis', 'Note', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('StatusModified', null, 'Statuszeitpunkt', 'Status time',
'u');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('AppOnDesktop', null, 'Desktop Symbol', 'Desktop icon',
'b');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('AutoStart', null, 'Autostart', 'Autostart', 'b');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_AppType', 'CALC_AppTypes.AppTypeID=AppTypeID,AppType',
'Typ', 'Type', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_Browser_CallPage', 'Browser-
.ApplicationID=ApplicationID,CallPage', 'Browser Seite', 'Browser
```

```
Page', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_FQName', 'CALC_FQName', 'Name mit Pfad', 'Name and
path', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_Groupname', 'Groups.GroupID=GroupID,Name', 'Organ-
isationseinheit', 'Organisation unit', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_ICA_Application', 'ICA.Ap-
plicationID=ApplicationID,Application', 'ICA Anwendung', 'ICA
Application', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_ICA_Server', 'ICA.ApplicationID=ApplicationID,Server',
'ICA Server', 'ICA Server', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_PNA_Server', 'PNA.Ap-
plicationID=ApplicationID,ServerUrl', 'PNA Server', 'PNA Server',
's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_RDP_Application', 'RDP.Ap-
plicationID=ApplicationID,Application', 'RDP Anwendung', 'RDP
Application', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_RDP_Server', 'RDP.ApplicationID=ApplicationID,Server',
'RDP Server', 'RDP Server', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_VD_Server', 'VDA.ApplicationID=ApplicationID,Server',
'VD Server', 'VD Server', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Name', null, null, null, 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('DisplayName', null, 'Anzeigename', 'Display name', 's');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('ReservedInt1', null, 'Autostart nach', 'Autostart delay',
'i');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Roaming', null, null, null, 'b');
INSERT INTO RptDefApp (Field, Access, Text_DE, Text_EN, Type)
VALUES ('TM', null, 'Dauerbetrieb', 'Restart', 'b');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('AssetName', null, 'Name', 'Name', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_AssetValue', 'CALC_AssetValue', 'Typ', 'Type', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('DeviceID', null, 'Geräte ID', 'Device id', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
```

```
VALUES ('EX_ProductID', 'CALC_ProductID', 'Produkt ID', 'Product
id', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_VendorID', 'CALC_VendorID', 'Hersteller ID', 'Vendor
id', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_ProductText', 'CALC_ProductText', 'Beschreibung',
'Description', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_VendorText', 'CALC_VendorText', 'Hersteller', 'Vendor',
's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_Serial', 'CALC_Serial', 'Seriennummer', 'Serial num-
ber', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_DeviceName', 'Device.DeviceID=DeviceID,Name', 'Ger-
ätename', 'Device name', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_DeviceMAC', 'Device.DeviceID=DeviceID,Mac_Address',
'MAC Adresse', 'MAC address', 's');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_FQName', 'CALC_FQName', 'Gerätename mit Pfad', 'Device
name with path', 's')
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('ClientIdentifier' ,NULL, 'Client Identifier','Client Iden-
tifier','s');
INSERT INTO RptDefAsset(Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_DeviceClientIdentifier', 'Device.DeviceID-
D=DeviceID,ClientIdentifier', 'Client-Identifier', 'Client-Iden-
tifier', 's');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('DeliveryNotificationID', null, 'Image-Ausliefer-
ungsvormerkung', 'Image delivery notification', 'i1');
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_TenantName', 'CALC_TenantName', 'Tenantname', 'Tenant
name', 's'); /*New field in report definition to display the tenant
name of a device */
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('DeliveryNotificationID', 'CALC_DeliveryNotification',
'Image-Auslieferungsvormerkung', 'Image delivery notification',
's'); /*New field in report definition to display the deliv-
erynotification state*/
INSERT INTO RptDefDevice (Field, Access, Text_DE, Text_EN, Type)
VALUES ('PeerIPAddress', null, 'Öffentliche Adresse', 'Public
address', 's'); /*Definition in report table to filter for the
```

```
public address*/
```

7.22. Table: RptDefGroup

This table contains all database fields of an ou which should be visible to the Scout Enterprise report tool.

ColumnName	DataType	Key	Not Null	Default Value	Comment
Field	NVARCHAR(255)	PK	NN		
Access	NVARCHAR(255)				
Text_DE	NVARCHAR(255)				
Text_EN	NVARCHAR(255)				
Type	NVARCHAR(255)				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	Field

HOW TO CREATE

```
CREATE TABLE RptDefGroup(
Field NVARCHAR(255) NOT NULL,
Access NVARCHAR(255) NULL,
Text_DE NVARCHAR(255) NULL,
Text_EN NVARCHAR(255) NULL,
Type NVARCHAR(255) NULL,
PRIMARY KEY (Field));
```

HOW TO INITIALIZE

```
INSERT INTO RptDefGroup (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_FQName', 'CALC_FQName', 'Name mit Pfad', 'Name and
path', 's');
INSERT INTO RptDefGroup (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_Location', 'Groups.GroupID=LocationID,Name', 'Organ-
isationseinheit', 'Organisation unit', 's');
INSERT INTO RptDefGroup (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Info1', null, null, null, 's');
INSERT INTO RptDefGroup (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Info2', null, null, null, 's');
INSERT INTO RptDefGroup (Field, Access, Text_DE, Text_EN, Type)
```

```
VALUES ('Info3', null, null, null, 's');
INSERT INTO RptDefGroup (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Name', null, null, null, 's');
INSERT INTO RptDefGroup (Field, Access, Text_DE, Text_EN, Type)
VALUES ('DeviceCount', null, 'Geräteanzahl', 'Device count', 'i');
INSERT INTO RptDefGroup (Field, Access, Text_DE, Text_EN, Type)
VALUES ('EX_DeviceCountSingle', 'CALC_DeviceCountSingle', 'Ger-
äteanzahl(nur OU)', 'Device count(only OU)', 'i');
INSERT INTO RptDefGroup (Field, Access, Text_DE, Text_EN, Type)
VALUES ('GroupID', null, 'OU-ID', 'OU-ID', 'i');
INSERT INTO RptDefGroup (Field, Access, Text_DE, Text_EN, Type)
VALUES ('Note', null, 'Hinweis', 'Note', 's');
```

7.23. Table: ScoutLicense

New license management Scout Enterprise 15

ColumnName	DataType	Key	Not Null	Default Value	Comment
ScoutLicenseID	INTEGER	PK	NN		Unique ID
LicenseBlob	VARBINARY (max)				
Modified	DATETIME				
ControlGuid	Uniqueidentifier				
AutoAssignLicenses	BIT		NN	1	Auto-assign application licenses 1: Application licenses are auto-assigned to requesting clients 0: Application licenses must be assigned by admin to be sent to clients

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ScoutLicenseID

HOW TO CREATE

```
CREATE TABLE License (
 ScoutLicenseID INTEGER NOT NULL,
 LicenseBlob VARBINARY(max),
 Modified DATETIME,
 ControlGuid Uniqueidentifier,
 AutoAssignLicenses BIT NOT NULL DEFAULT 1
 CONSTRAINT PK_ScoutLicense,
 PRIMARY KEY (ScoutLicenseID));
```

HOW TO INITIALIZE

```
INSERT INTO ScoutLicense (ScoutLi-
censeID,LicenseBlob,Modified,ControlGuid) VALUES
(0,0x8886726839,GETUTCDATE(),NEWID());
```

7.24. Table: ServerAction

Server actions are a integral part of the internal communications between Scout Enterprise components. Table entries should never be manually added, removed or changed.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ServerActionID	INTEGER IDENTITY (1,1)	PK	NN		Unique ID of server action entry
Type	INTEGER		NN		
ServerID	INTEGER	(FK)	NN		Target server of the action Refers to EntityID of table Entity
StartTime	INTEGER		NN		
MemberType	INTEGER				
MemberID	INTEGER				
RequestingConsole	INTEGER				
StringParameter	NVARCHAR(255)				
IntParameter	INTEGER				
StringParameter2	NVARCHAR(255)				
IntParameter2	INTEGER				
StringParameter3	NVARCHAR(255)				
IntParameter3	INTEGER				
StringParameter4	NVARCHAR(255)				
IntParameter4	INTEGER				
StringParameter5	NVARCHAR(255)				
IntParameter5	INTEGER				
LongStringParameter	NVARCHAR(max)				
ReservedString1	NVARCHAR(255)				
ReservedString2	NVARCHAR(255)				

ColumnName	DataType	Key	Not Null	Default Value	Comment
ReservedInt1	INTEGER				
ReservedInt2	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ServerActionID

HOW TO CREATE

```
CREATE TABLE ServerAction(
 ServerActionID INTEGER NOT NULL,
 Type INTEGER NOT NULL,
 ServerID INTEGER,
 StartTime INTEGER NOT NULL,
 MemberType INTEGER,
 MemberID INTEGER,
 RequestingConsole INTEGER,
 StringParameter NVARCHAR(255),
 IntParameter INTEGER,
 StringParameter2 NVARCHAR(255),
 IntParameter2 INTEGER,
 StringParameter3 NVARCHAR(255),
 IntParameter3 INTEGER,
 StringParameter4 NVARCHAR(255),
 IntParameter4 INTEGER,
 StringParameter5 NVARCHAR(255),
 IntParameter5 INTEGER,
 LongStringParameter NVARCHAR(max),
 ReservedString1 NVARCHAR(255),
 ReservedString2 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 PRIMARY KEY (ServerActionID));
```

7.25. Table: ServerOptions

Server options are stored as a combination of parameter name and value. Changes should only be done by using the Scout console program.

ColumnName	DataType	Key	Not Null	Comment
ServerOptionsID	INTEGER	PK	NN	Unique ID of server option entry
ServerID	INTEGER		NN	
ParamName	NVARCHAR(255)		NN	Pre-defined name of parameter:
ParamVal	NVARCHAR(255)			Parameter depending value (see ParamName)
ParamIntVal	INTEGER			

Name	Values	Def	Description
UseClassicLogin	0|1	1	Indicates whether classic Scout logon (administrator and guest only) or administrator policies are used
OUFilter	0|1|2	0	Type of OU filter that is activated 0: No filter 1: Subnet filter 2: Custom filter
OUFilterIgnoreDefault	0|1	1	If value is 1 all devices that have not matched filter criteria are not moved to the default OU

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ServerOptionsID

HOW TO CREATE

```
CREATE TABLE ServerOptions(
 ServerOptionsID INTEGER NOT NULL,
 ServerID INTEGER NOT NULL,
 ParamName NVARCHAR(255) NOT NULL,
 ParamVal NVARCHAR(255),
 ParamIntVal INTEGER,
 PRIMARY KEY (ServerOptionsID));
```

HOW TO INITIALIZE

```
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(1,1,'DebugLevel','',1);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(2,1,'Password','_]pZWIeXe',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(3,1,'AllowDynamicHostnames','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(4,1,'WOLWithIpAddress','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
```

```
(5,1,'WOLWithUDP','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(6,1,'AllowDynamicGroupID','',1);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(7,1,'CheckNetCrossing','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(8,1,'ManageOnlyLockedFields','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(9,1,'RenameExistingDevice','',1);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(10,1,'DeactivateNewDevices','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(11,1,'DoSmartSrv','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(12,1,'SmartSrvTimeout','',30);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(13,1,'ScoutSrvGroup','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(14,1,'DiscoverPingTime','',20);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(15,1,'DiscoverCollectTime','',30);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(16,1,'WolServer','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(17,1,'PrinterImportTime','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(18,1,'UpdateDelay','',30);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
```

```
(19,1,'UpdateCheck','',10);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(20,1,'UpdateTimeout','',360);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(21,1,'UpdateMaxAtOnce','',10);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(22,1,'UpdateMaxStartDuration','',300);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(23,1,'UpdateConnectTimeout','',2);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(24,1,'DiscoverOnStart','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(25,1,'RecoverURL','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(26,1,'RecoverProxy','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(27,1,'RecoverPort','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(28,1,'CmdHalt','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(29,1,'CmdUpdate','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(30,1,'CmdReboot','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(31,1,'CmdRestartX','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(32,1,'CmdMsg','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
```

```
(33,1,'UseClassicLogin','',1);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(34,1,'MonitoringActive','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(35,1,'LastApplicationAction','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(36,1,'LastApplicationActionDelay','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(37,1,'UsePeerIP','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(38,1,'FirstContactAction','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(39,1,'MinRootPartitionSize','',128);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(40,1,'MaxRootPartitionSize','',3072);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(41,1,'SwapPartitionSize','',2);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(42,1,'RootPartitionSize','',128);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(43,1,'HomePartitionSize','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(44,1,'MinHomePartitionSize','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(45,1,'MaxHomePartitionSize','',4096);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(46,1,'RecoverAskUser','',1);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
```

```
(47,1,'RecoverUsePartitionData','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(48,1,'MaxDBDataSize','',500);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(49,1,'RecoverRL','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(50,1,'RecoverRLMenu','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(51,1,'SubscriptionWarning','',3);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(52,1,'ServerDataDir','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(53,1,'MirrorInvitationTimeout','',20);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(54,1,'DevNameTemplate','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(55,1,'OUFilter','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(56,1,'OUFilterIgnoreDefault','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(57,1,'AcceptDevices','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(58,1,'UseConfigSnapshot','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(59,1,'UseReportAdmins','',0);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(60,1,'CheckDCGPermissions','',1);
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
```

```
(61,1,'SDIValidity','',300); /*Validity period of simple device identifiers*/
INSERT INTO ServerOptions (ServerOptionsID, ServerID , ParamName, ParamVal, ParamIntVal) VALUES
(62,1,'SDIInfo','',0) /*Text which can be displayed on the client when a simple device identifier is
requested */
```

7.26. Table: ServiceProviderReport

ColumnName	DataType	Key	Not Null	Default Value	Comment
ServiceProviderReportID	UNIQUEIDENTIFIER	PK	NN		Unique ID of service provider report
ReportCreated	DATETIME		NN		
ReportSent	DATETIME	(FK)	NN		
ReportText	NVARCHAR(max)		NN		
ReportFile	NVARCHAR(max)		NN		
ReportSentOk	BIT		NN		

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ServiceProviderReportID

HOW TO CREATE

```
CREATE TABLE ServiceProviderReport(
 ServiceProviderReportID UNIQUEIDENTIFIER NOT NULL,
 ReportCreated DATETIME NOT NULL,
 ReportSent DATETIME NOT NULL,
 ReportText NVARCHAR(max)NOT NULL,
 ReportFile NVARCHAR(max)NOT NULL,
 ReportSentOk BIT NOT NULL,
 CONSTRAINT PK_ServiceProviderReport
 PRIMARY KEY (ServiceProviderReportID));
```

7.27. Table: System

System properties are stored as a combination of parameter name and value. It is highly recommended not to change any entry.

ColumnName	DataType	Key	Not Null	Default Value	Comment
SystemID	INTEGER	PK	NN		
ParamName	NVARCHAR(255)		NN		
ParamVal	NVARCHAR(255)		NN		

IndexName	IndexType	Columns
PRIMARY	PRIMARY	SystemID

HOW TO CREATE

```
CREATE TABLE System(
 SystemID INTEGER NOT NULL,
 ParamName NVARCHAR(255) NOT NULL,
 ParamVal NVARCHAR(255) NOT NULL,
 PRIMARY KEY (SystemID));
```

7.28. Table: UCFT

This table stores all binary configuration data such as imported file entries, imported background images and application icons.

ColumnName	DataType	Key	Not Null	Default Value	Comment
UCFTID	INTEGER	PK	NN		
MemberType	INTEGER				
MemberID	INTEGER				
FileType	INTEGER				
FileMode	INTEGER				
FileContent	VARBINARY(max)				
FileName	NVARCHAR(255)				
FTResult	INTEGER				
TimeModified	INTEGER				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	UCFTID

HOW TO CREATE

```
CREATE TABLE UCFT(
UCFTID INTEGER NOT NULL,
MemberType INTEGER,
MemberID INTEGER,
FileType INTEGER,
FileMode INTEGER,
FileContent VARBINARY(max),
FileName NVARCHAR(255),
FTResult INTEGER,
TimeModified INTEGER,
ReservedString1 NVARCHAR(255),
ReservedString2 NVARCHAR(255),
ReservedInt1 INTEGER,
ReservedInt2 INTEGER,
PRIMARY KEY (UCFTID));
```

7.29. Table: UDM

It is highly recommended not to change any entry.

ColumnName	DataType	Key	Not Null	Default Value	Comment
UDMID	INTEGER	PK	NN		
DeviceID	INTEGER		NN		
DeviceMessageID	INTEGER				
MessageText	NVARCHAR(255)				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	UDMID

HOW TO CREATE

```
CREATE TABLE UDM(
 UDMID INTEGER NOT NULL,
 DeviceID INTEGER NOT NULL,
 DeviceMessageID INTEGER,
 MessageText NVARCHAR(255),
 PRIMARY KEY (UDMID));
```

7.30. Table: UpdateNotification

Handles the Image update notification feature in version 14.9 and later

ColumnName	DataType	Key	Not Null	Default	Comment
UpdateNotificationID	INTEGER IDENTITY (1,1)	PK	NN		Unique ID of notification
[Type]	INTEGER		NN	0	Defines if a notification is an update or delivery notification
Options	INTEGER		NN		Bit field for update notification options SCD_OPTION_FORMAT_BEFORE_UPDATE 0x40; SCD_OPTION_USERINFO; 0x04 SCD_OPTION_CANREFUSE; 0x08
ConfirmTimeout	INTEGER		NN	0	
Modified	DATETIME		NN	GETUTCDATE ()	Internal use
ReservedString1	NVARCHAR (255)				
ReservedInt1	NVARCHAR (255)				
ReservedInt2	NVARCHAR (255)				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	UpdateNotificationID

HOW TO CREATE

```
CREATE TABLE ServerOptions(
 UpdateNotificationID INTEGER IDENTITY (1,1)NOT NULL DEFAULT(0),
 [Type] INTEGER NOT NULL DEFAULT(0),
 Options INTEGER NOT NULL,
 ConfirmTimeout INTEGER NOT NULL DEFAULT(0),
 Modified DATETIME NOT NULL DEFAULT(GETUTCDATE()),
 ReservedString1 NVARCHAR(255),
 ReservedInt1 INTEGER,
 ReservedInt2 INTEGER,
 CONSTRAINT PK_UpdateNotification
 PRIMARY KEY (UpdateNotificationID));
```

7.31. Table: ViewConnectedConsole

This is a view data table which is used by Scout Enterprise Dashboard. The table is filled automatically. It is highly recommended not to change any entry.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ViewConnectedConsoleId	INTEGER	PK	NN		
Source	INTEGER		NN		
Online	INTEGER		NN		
Modified	DATETIME		NN		

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ViewConnectedConsoleId

How to create

```
CREATE TABLE ViewConnectedConsole(
 ViewConnectedConsole INTEGER NOT NULL,
 Source INTEGER NOT NULL,
 Online INTEGER NOT NULL,
 Modified DATETIME NOT NULL,
 CONSTRAINT PK_ViewConnectedConsole
 PRIMARY KEY (ViewConnectedConsoleId));
```

7.32. Table: ViewDatabaseInformation

This is a view data table which is used by Scout Enterprise Dashboard. The table is filled automatically. It is highly recommended not to change any entry.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ViewDatabaseInformationID	INTEGER IDENTITY (1,1)	PK	NN		
Source	INTEGER		NN		
Name	NVARCHAR(255)		NN		
Instance	NVARCHAR(255)		NN		
Type	INTEGER		NN		
Hostname	NVARCHAR(255)		NN		
Version	NVARCHAR(10)		NN		
Modified	DATETIME		NN		

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ViewDatabaseInformationID

How to create

```
CREATE TABLE ViewDatabaseInformation(
 ViewDatabaseInformationID INTEGER IDENTITY (1,1) NOT NULL,
 Source INTEGER NOT NULL,
 Name NVARCHAR(255) NOT NULL,
 Instance NVARCHAR(255) NOT NULL,
 Type INTEGER NOT NULL,
 Hostname NVARCHAR(255) NOT NULL,
 Version NVARCHAR(10) NOT NULL,
 Modified DATETIME NOT NULL,
 CONSTRAINT PK_ViewDatabaseInformation
 PRIMARY KEY (ViewDatabaseInformationID));
```

7.33. Table: ViewDistributedImage

This is a view data table which is used by Scout Enterprise Dashboard. The table is filled automatically. It is highly recommended not to change any entry.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ViewDistributedImageID	INTEGER IDENTITY (1,1)	PK	NN		
ImageFile	NVARCHAR(255)		NN		
Outdated	BIT		NN		
InstalledImages	INTEGER		NN		
Modified	DATETIME		NN		

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ViewDistributedImageID

How to create

```
CREATE TABLE ViewDistributedImage(
 ViewDistributedImageID INTEGER IDENTITY (1,1) NOT NULL,
 ImageFile NVARCHAR(255) NOT NULL,
 Outdated BIT NOT NULL,
 InstalledImages INTEGER NOT NULL,
 Modified DATETIME NOT NULL,
 CONSTRAINT PK_ViewDistributedImage
 PRIMARY KEY (ViewDistributedImageID));
```

7.34. Table: ViewLicenseTermtime

This is a view data table which is used by Scout Enterprise Dashboard. The table is filled automatically. It is highly recommended not to change any entry.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ViewLicenseTermtimeID	INTEGER IDENTITY (1,1)	PK	NN		
SubscriptionMode	INTEGER		NN		
ValidUntil	DATETIME		NN		
Modified	DATETIME		NN		

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ViewLicenseTermtimeID

How to create

```
CREATE TABLE ViewLicenseTermtime(
 ViewLicenseTermtimeID INTEGER IDENTITY (1,1) NOT NULL,
 SubsriptionMode INTEGER NOT NULL,
 ValidUntil DATETIME NOT NULL,
 Modified DATETIME NOT NULL,
 CONSTRAINT PK_ViewLicenseTermtime
 PRIMARY KEY (ViewLicenseTermtimeID));
```

7.35. Table: ViewLicenseUsage

This is a view data table which is used by Scout Enterprise Dashboard. The table is filled automatically. It is highly recommended not to change any entry.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ViewLicenseUsageID	INTEGER IDENTITY (1,1)	PK	NN		
Name	NVARCHAR(255)		NN		
Type	INTEGER		NN		
Used	INTEGER		NN		
Available	INTEGER		NN		
Without	INTEGER		NN		
Modified	DATETIME		NN		

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ViewLicenseUsageID

How to create

```
CREATE TABLE ViewLicenseUsage(
 ViewLicenseUsageId INTEGER IDENTITY (1,1) NOT NULL,
 Name NVARCHAR(255) NOT NULL,
 Type INTEGER NOT NULL, Used INTEGER NOT NULL,
 Available INTEGER NOT NULL,
 Without INTEGER NOT NULL,
 Modified DATETIME NOT NULL,
 CONSTRAINT PK_ViewLicenseUsage
 PRIMARY KEY (ViewLicenseUsageId));
```

7.36. Table: ViewManagedThinClientStatus

This is a view data table which is used by Scout Enterprise Dashboard. The table is filled automatically. It is highly recommended not to change any entry.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ViewManagedThinClientStatusID	INTEGER IDENTITY (1,1)	PK	NN		
Online	INTEGER		NN		
Offline	INTEGER		NN		
Modified	DATETIME		NN		

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ViewManagedThinClientStatusID

How to create

```
CREATE TABLE ViewManagedThinClientStatus(
 ViewManagedThinClientStatusID INTEGER IDENTITY (1,1) NOT NULL,
 Online INTEGER NOT NULL,
 Offline INTEGER NOT NULL,
 Modified DATETIME NOT NULL,
 CONSTRAINT PK_ViewThinClientStatus PRIMARY KEY (ViewMan-
agedThinClientStatusID));
```

7.37. Table: ViewServerInformation

This is a view data table which is used by Scout Enterprise Dashboard. The table is filled automatically. It is highly recommended not to change any entry.

ColumnName	DataType	Key	Not Null	Default Value	Comment
ViewServerInformationID	INTEGER IDENTITY (1,1)	PK	NN		
Name	NVARCHAR(255)		NN		
Type	INTEGER		NN		
Uptime	DATETIME		NN		
Online	BIT		NN		
Modified	DATETIME		NN		

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ViewConnectedConsoleId

How to create

```
CREATE TABLE ViewServerInformation(
 ViewServerInformationID INTEGER IDENTITY (1,1) NOT NULL,
 Name NVARCHAR(255) NOT NULL, Type INTEGER NOT NULL,
 Uptime DATETIME NOT NULL,
 Online BIT NOT NULL,
 Modified DATETIME NOT NULL,
 CONSTRAINT PK_iewServerInformation
 PRIMARY KEY (ViewServerInformationID))
```