Johnson Controls

P2000

Security Management System

Database Table Definitions

for Creating Custom Reports

Version 3.11, October, 2011
24-10618-7 Revision –

P2000

Security Management System

Database Table Definitions

for Creating Custom Reports

Version 3.11, October, 2011

24-10618-7 Revision –

Johnson Controls
Security Solutions
(805) 522-5555
www.johnsoncontrols.com

Acknowledgment

DATABASE TABLE DEFINITIONS

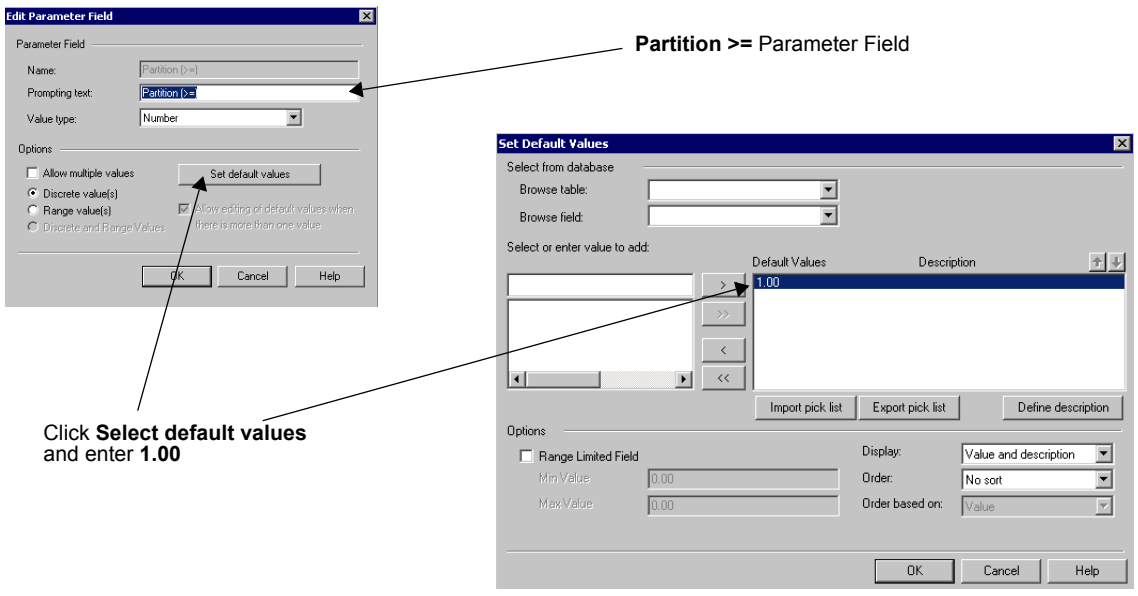
OVERVIEW

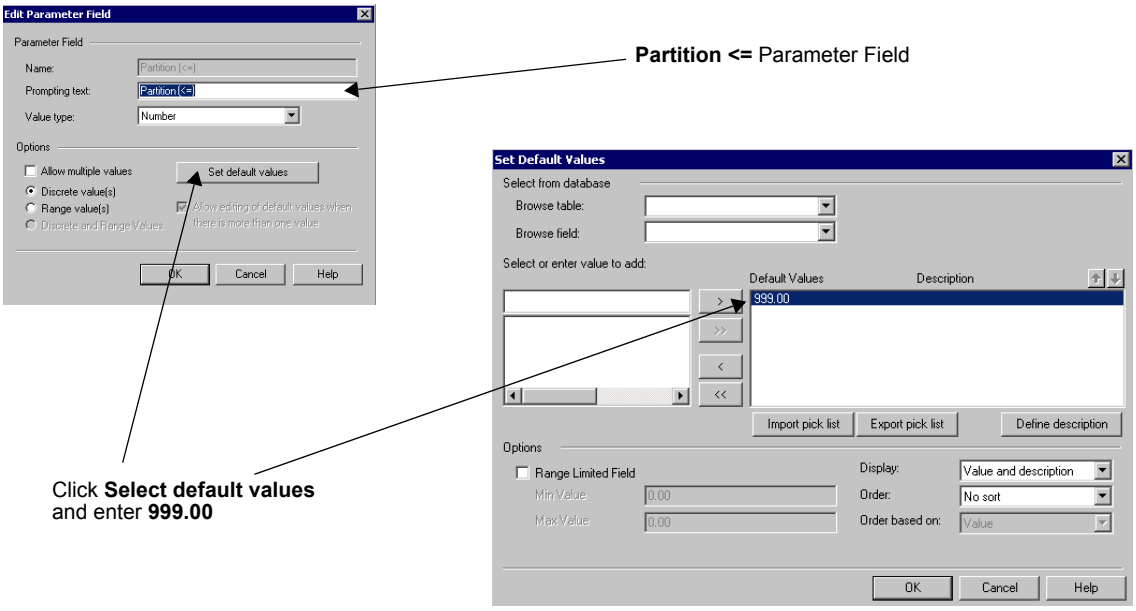
The P2000 Database Table Definitions are provided to assist you in creating custom reports using SAP® Crystal Reports®. You can create custom reports using SAP Crystal Reports and then import them into the P2000 system. You can also export a P2000 standard report, open it in SAP Crystal Reports, edit it, and then import it back into the P2000 system. This document assumes you have your own copy of Crystal Reports, and you are familiar on how reports are run by the P2000 system.

To write custom reports you need to know how the database is constructed. You must know the field/table relationships for the information you need before you can create new fields for your reports. This document contains a description of all the database tables used in the P2000 system and a brief description of what each field contains. Tables often contain a unique identifier field that is generated by the database engine as items are added. This identifier is used to relate entries in the database to each other. For example, all terminals installed on a panel with a *p_panel_id* of *1000* will have their *tp_panel_id* field also set to a value of *1000*, this allows a report to print a list of all terminals on a specific panel.

In a partitioned system, custom reports created for the P2000 system must include two parameter fields, “Partition (>=)” and “Partition (<=)”, these will be supplied by the P2000 system when the report is launched. These parameters must be used to filter the report data so only information for the correct partition(s) appears on the report.

The following examples show how the Partition parameter fields are defined. These dialog boxes may differ slightly depending on the Crystal Reports version you are using.





When creating your reports, be sure to use the Pegasys ODBC Data Source as your database connection. Using another data source may prevent the report from running correctly from P2000 workstations. The Pegasys ODBC Data Source is created by the P2000 installation.

NOTE

Custom reports may need to be modified after a P2000 software version upgrade, since the database structure may have changed.

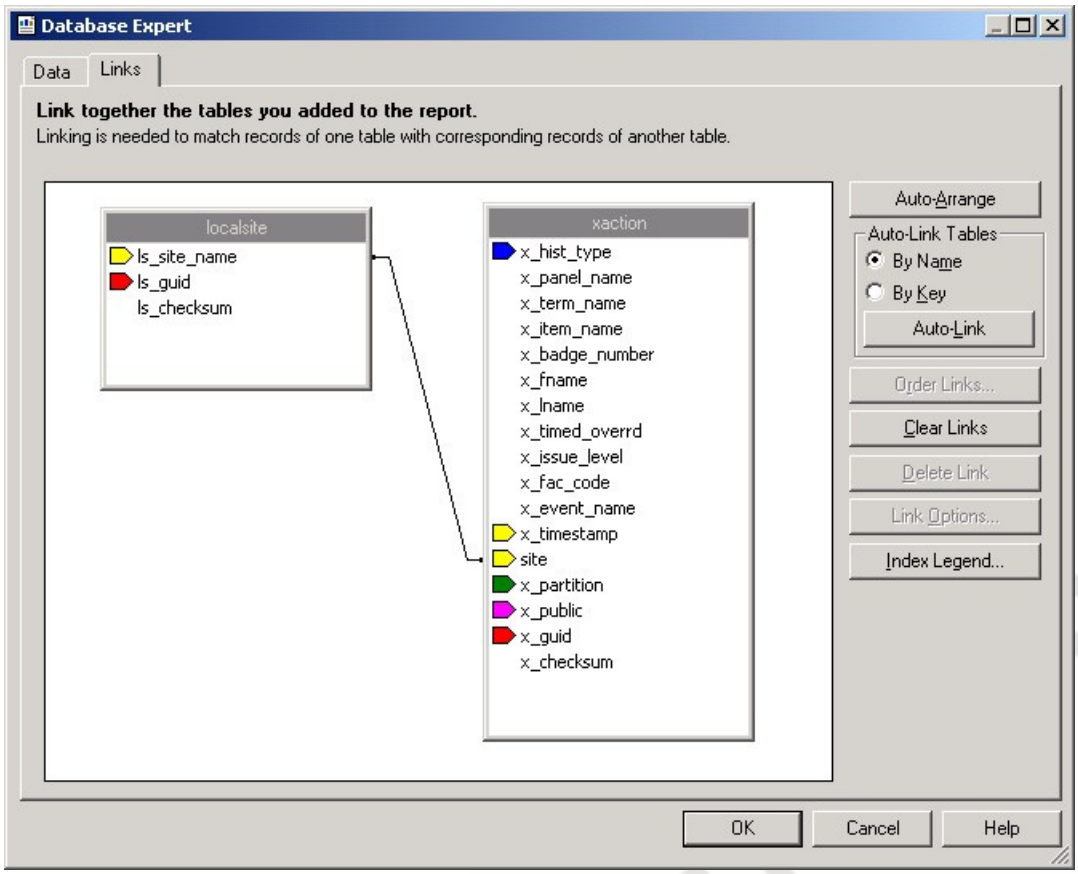
Refer to the field/table relationship information starting on page 8 to create your reports, and follow the instructions provided in your Crystal Reports documentation.

P2000 VER. 3.0 (AND HIGHER) IMPACT ON CUSTOM REPORTS

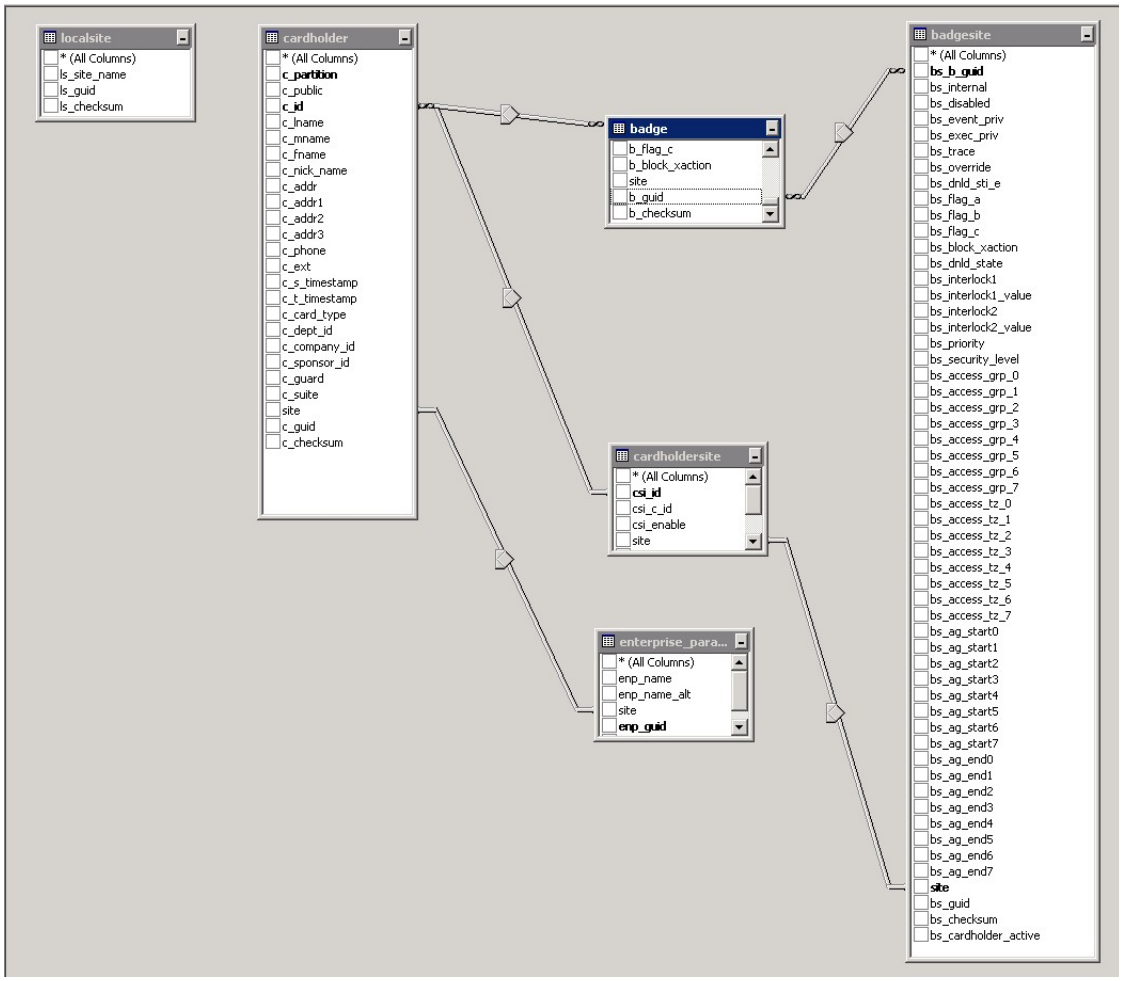
P2000 version 3.0 introduced the P2000 Enterprise option, which allows customers with multiple sites to communicate with each other to share Cardholder/Badge information. Cardholders can be granted access to doors at all assigned sites within the Enterprise system.

In order to implement the centralized cardholder management for P2000 Enterprise systems, the existing cardholder/badge relationships and tables had to be modified. Accordingly, new rules must be applied to both new and existing Custom Reports implemented on any P2000 system that reports upon cardholder/badge records.

In addition, P2000 Enterprise users should note that even if no custom reporting is done against cardholder/badge records, the local site table must be added to the report, and linked by the site name to the corresponding site field on the respective tables. This is necessary in order to present data specific only to the site that the report is being run on. If however, it is desired that records for all sites be reported on, then linking to the **localsite** table is not necessary. An example of such linking against P2000 Transaction Records on a P2000 Enterprise system is demonstrated below:



The following diagram outlines the Entity Relationship Diagram (ERD) for cardholder and badge tables:



The tables **cardholder** and **badge** contain information that is applicable for the entire P2000 Enterprise system, where the tables **cardholdersite** and **badgesite** contain site-specific information. On a P2000 system with no Enterprise option, the tables **cardholdersite** and **badgesite** will contain the P2000 system settings.

The following table outlines the database columns that have been moved or require a different interpretation:

P2000 V2.5		P2000 V3.0 and higher		
Table	**Column**	**Table**	**Column**	**Contents**
Cardholder	c_cardholder_active	Cardholdersite	csi_active	1 – Cardholder active on site **cardholdersite.site** 0 – Cardholder not active on site **cardholdersite.site**
		Cardholdersite	csi_enable	1 – Cardholder enabled on site **cardholdersite.site** 0 – Cardholder disabled
Badge	b_cardholder_active	BadgeSite	bs_cardholder_active	1 – Cardholder active on site **badgesite.site** 0 – Cardholder not active on site **badgesite.site**
Badge	b_access_grp_x	Badge	b_access_grp_x	Enterprise Access Group x
Badge	b_timezone_x	Badge	b_timezone_x	Enterprise Timezone x

Badge	b_exec_priv	Badge/BadgeSite	b_exec_priv
			bs_exec_priv
				If (IsNull(bs_internal,1) = 1) then
				Use b_exec_priv
				Else
				Use bs_exec_priv
				1 – Executive privilege granted

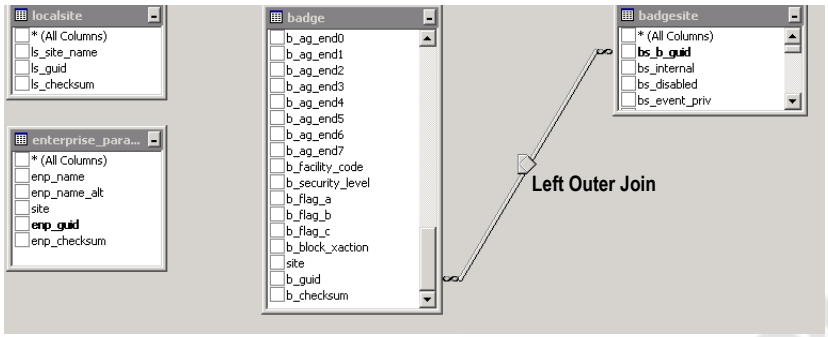
Badge	b_trace	Badge/BadgeSite	b_trace
			bs_trace
				If (IsNull(bs_internal) = 1) then
				Use b_trace
				Else
				Use bs_trace
				1 – Trace badge messages

Enterprise Access Groups do not contain any references to terminals, terminals groups or other hardware items. An Enterprise Access Group links to a local Access Group by name only.

Recommended Table Joins

We recommend using the following table joins for linking the different tables:

(a) Badge and Badge Site

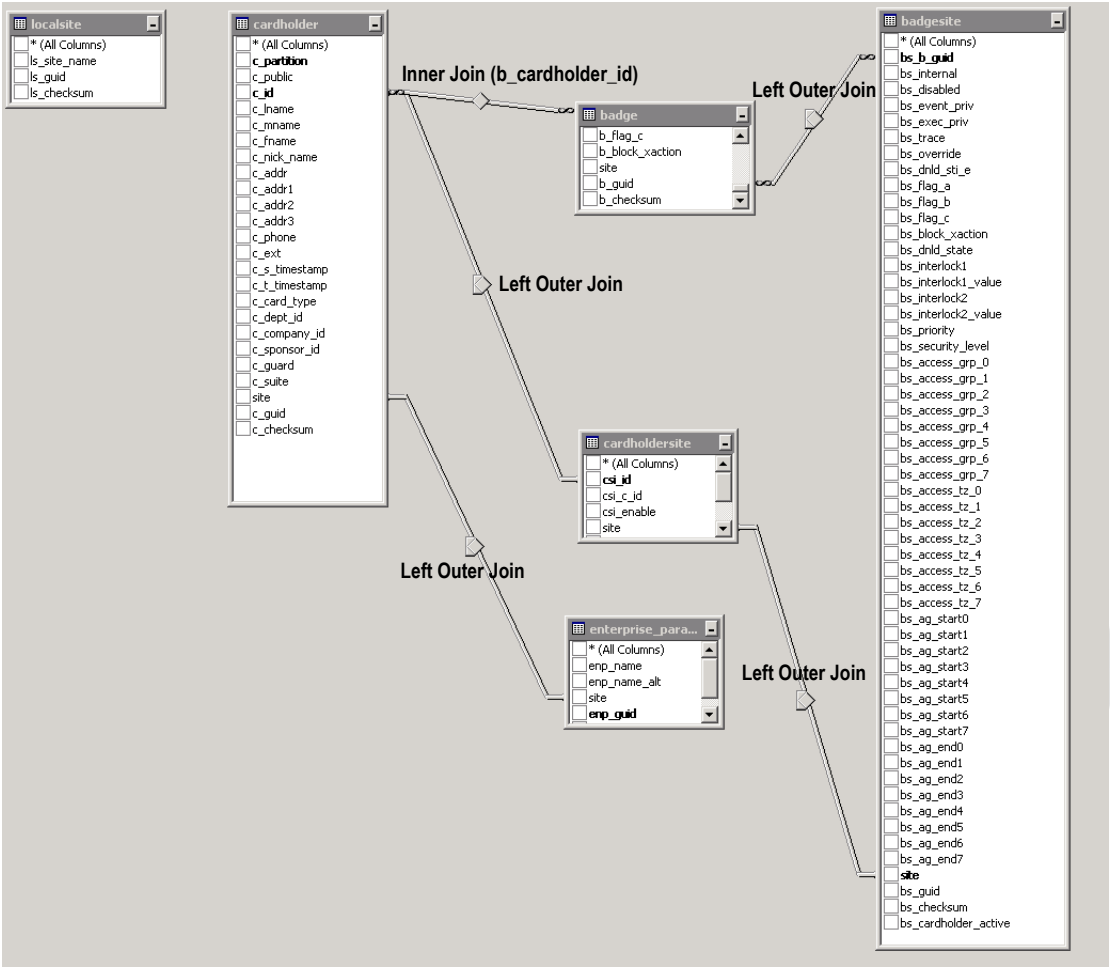


You can use the following selection formula in Crystal Reports to apply the Enterprise rules:

```
if isnull({enterprise_parameters.enp_name})  then
        (
        // Local Site, ie no Enterprise
        if IsNull({badgesite.site}) then
                FALSE
        else
        if ({badgesite.site} <> {localsite.ls_site_name}) then
```

```
                FALSE
        else
                TRUE
        )
else
        (
        if ({enterprise_parameters.enp_name} <> {localsite.ls_site_name}) and
        ({badgesite.site} <> {localsite.ls_site_name}) then
        // Not central site  and
        // Badge site not local site
                FALSE
        else
        // We are running on the central site or on local site with local records
                TRUE
        )
```

(b) Cardholder, Cardholder Site, Badge and BadgeSite



You can use the following selection formula in Crystal Reports to apply the Enterprise rules:

```
if isnull({enterprise_parameters.enp_name})  then
(
        // Local Site, ie no Enterprise
        if IsNull({badgesite.site}) then
```

```
                FALSE
        else
        if ({badgesite.site} <> {localsite.ls_site_name}) then
                FALSE
        else
                TRUE
)
else
(
        if {cardholdersite.csi_enable} = 0 then
                FALSE
        else
        if ({enterprise_parameters.enp_name} <> {localsite.ls_site_name}) and
          ({cardholdersite.site} <> {localsite.ls_site_name}) then
                // Not central site  and
                // Cardholder site not local site
                FALSE
        else
        (
                // We are running on the central site or on local site with local records
                TRUE
        )
)
```

When coding reports for a P2000 Enterprise system, any Crystal Reports filtering and table joins should consider that a local **badgesite** record might not exist.

In order to update existing custom reports please note that the following steps must take place after P2000 is upgraded to version 3.0 (or higher):

1. Using the Crystal Reports Designer component, you must verify the database by executing the **Verify Database** function located under the **Database** menu item on the Crystal Reports Designer toolbar.
2. If reporting on Cardholder/Badge records, the table structures outlined in this section must be incorporated by adding in and linking the appropriate table. The **Selection Formula** for the report must also be updated to incorporate the necessary filters as outlined in this section.
3. If reporting on site-specific data that is not related to Cardholder or Badge records on a P2000 Enterprise system, where records must be displayed only for the local site that the report is being run on, then the **localsite** table must be added to the report. In addition, the field **ls_site_name** must be linked to the corresponding site field on the top level table already existing in the report. An example of such table link is detailed at the beginning of this section.

   It is important that if the **localsite** table is linked in the report, a field from the **localsite** table must be utilized on the report, otherwise, by default, Crystal Reports will ignore the link to the **localsite** table. This can be done by including a field from the **localsite** table on the report, and then suppressing the field.

CREATE TABLE accgroup

```
(
  ag_partition          INTEGER REFERENCES partition(part_number)
                            NOT FOR REPLICATION,          -- partition ID to which this cardholder belongs
  ag_public             SMALLINT,                         -- 1 = access group is public, 0 = access group is private
  ag_id                 INTEGER IDENTITY
                            NOT FOR REPLICATION UNIQUE,   -- unique database ID
  ag_name               CHAR(32) NOT NULL,                -- access group name
  ag_enable             SMALLINT,                         -- 1 = enabled, - = disabled
  ag_default_tz         INTEGER REFERENCES timezone(tz_id)
                            NOT FOR REPLICATION,          -- default timezone which this cardholder has
  site                  CHAR(32) NOT NULL,                -- Site Name
  ag_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  ag_checksum           INTEGER NOT NULL,                 -- Record checksum
  PRIMARY KEY           ( ag_partition, ag_id )
);
  CREATE INDEX              ag_public ON accgroup ( ag_public );
  CREATE UNIQUE INDEX       ag_site_name ON accgroup (site, ag_name);
```

CREATE TABLE access_group_details

```
(
  cag_ag_id             INTEGER REFERENCES accgroup(ag_id)
                            NOT FOR REPLICATION,
  cag_term_id           INTEGER REFERENCES terminal(tp_term_id)
                            NOT FOR REPLICATION,          -- terminal ID
  cag_tz                INTEGER REFERENCES timezone(tz_id)
                            NOT FOR REPLICATION,
  cag_default           SMALLINT,                         -- is default value or not
  site                  CHAR(32) NOT NULL,                -- Site Name
  cag_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  cag_checksum          INTEGER NOT NULL,                 -- Record checksum
  PRIMARY KEY           ( cag_ag_id, cag_term_id )
);
```

CREATE TABLE access_group_details_save

```
(
  cag_ag_id             INTEGER NOT NULL,
  cag_term_id           INTEGER NOT NULL,                 -- terminal ID
  cag_tz                INTEGER,
  cag_default           SMALLINT,                         -- is default value or not
  site                  CHAR(32) NOT NULL,                -- Site Name
  cag_guid              uniqueidentifier NOT NULL,        -- Global unique record identifier
  cag_checksum          INTEGER NOT NULL,                 -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,        -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                 -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                            newid() UNIQUE NOT NULL,      -- Global unique record identifier
  PRIMARY KEY           ( cag_ag_id, cag_term_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE access_group_number

```
(
  agn_ag_id             INTEGER REFERENCES accgroup(ag_id)
                            NOT FOR REPLICATION,
  agn_panel_id          INTEGER REFERENCES panel(p_panel_id)
                            NOT FOR REPLICATION,          -- panel ID
  agn_al_number         SMALLINT,
  agn_checksum          INTEGER NOT NULL,                 -- Record checksum
  PRIMARY KEY           ( agn_ag_id, agn_panel_id, agn_al_number)
);
```

CREATE TABLE accgroup_save

```
(
  ag_partition          INTEGER NOT NULL,                          -- partition ID to which this cardholder belongs
  ag_public             SMALLINT,                                  -- 1 = access group is public, 0 = access group is private
  ag_id                 INTEGER NOT NULL,                          -- unique database ID
  ag_name               CHAR(32) NOT NULL,                         -- access group name
  ag_enable             SMALLINT,                                  -- 1 = enabled, - = disabled
  ag_default_tz         INTEGER,                                   -- default timezone which this cardholder has
  site                  CHAR(32) NOT NULL,                         -- Site Name
  ag_guid               uniqueidentifier NOT NULL,                 -- Global unique record identifier
  ag_checksum           INTEGER NOT NULL,                          -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                 -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                         -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                          -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                            newid() UNIQUE NOT NULL,               -- Global unique record identifier
  PRIMARY KEY           ( ag_partition, ag_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE accgrpelevator

```
(
  age_ag_id             INTEGER REFERENCES accgroup(ag_id)
                            NOT FOR REPLICATION,                   -- access group ID
  age_grp_id            INTEGER REFERENCES floorgrp(fg_id)
                            NOT FOR REPLICATION,                   -- floor/door group ID
  site                  CHAR(32) NOT NULL,                         -- Site Name
  age_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  age_checksum          INTEGER NOT NULL,                          -- Record checksum
  PRIMARY KEY           ( age_ag_id, age_grp_id)
);
```

CREATE TABLE accgrpelevator_save

```
(
  age_ag_id             INTEGER NOT NULL,                          -- access group ID
  age_grp_id            INTEGER NOT NULL,                          -- floor/door group ID
  site                  CHAR(32) NOT NULL,                         -- Site Name
  age_guid              uniqueidentifier NOT NULL,                 -- Global unique record identifier
  age_checksum          INTEGER NOT NULL,                          -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                 -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                         -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                          -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                            newid() UNIQUE NOT NULL,               -- Global unique record identifier
  PRIMARY KEY           ( age_ag_id, age_grp_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE accgrpterm

```
(
  agt_ag_id             INTEGER REFERENCES accgroup(ag_id)
                            NOT FOR REPLICATION,                   -- access group ID
  agt_term_id           INTEGER REFERENCES terminal(tp_term_id)
                            NOT FOR REPLICATION,                   -- terminal ID
  site                  CHAR(32) NOT NULL,                         -- Site Name
  agt_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  agt_checksum          INTEGER NOT NULL,                          -- Record checksum
  PRIMARY KEY           ( agt_ag_id, agt_term_id)
);
  CREATE NONCLUSTERED INDEX     agt_ag_id ON accgrpterm(agt_ag_id ASC);
```

CREATE TABLE accgrptermgrp

```
(
  agtg_ag_id            INTEGER REFERENCES accgroup(ag_id)
                            NOT FOR REPLICATION,                          -- access group ID
  agtg_termgrp_id       INTEGER REFERENCES termgrp(tg_id)
                            NOT FOR REPLICATION,                          -- terminal group ID
  site                  CHAR(32) NOT NULL,                              -- Site Name
  agtg_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
  agtg_checksum         INTEGER NOT NULL,                               -- Record checksum
  PRIMARY KEY           ( agtg_ag_id, agtg_termgrp_id)
);
  CREATE NONCLUSTERED INDEX      agtg_ag_id ON accgrptermgrp(agtg_ag_id ASC);
  CREATE NONCLUSTERED INDEX      agtg_termgrp_id ON accgrptermgrp(agtg_termgrp_id ASC);
```

CREATE TABLE accgrptermgrp_save

```
(
  agtg_ag_id            INTEGER NOT NULL,                               -- access group ID
  agtg_termgrp_id       INTEGER NOT NULL,                               -- terminal group ID
  site                  CHAR(32) NOT NULL,                              -- Site Name
  agtg_guid             uniqueidentifier NOT NULL,                      -- Global unique record identifier
  agtg_checksum         INTEGER NOT NULL,                               -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                      -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                              -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                              -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                               -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL
                            DEFAULT newid() UNIQUE NOT NULL,            -- Global unique record identifier
  PRIMARY KEY           ( agtg_ag_id, agtg_termgrp_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE accgrpterm_save

```
(
  agt_ag_id             INTEGER NOT NULL,                               -- access group ID
  agt_term_id           INTEGER NOT NULL,                               -- terminal ID
  site                  CHAR(32) NOT NULL,                              -- Site Name
  agt_guid              uniqueidentifier NOT NULL,                      -- Global unique record identifier
  agt_checksum          INTEGER NOT NULL,                               -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                      -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                              -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                              -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                               -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                            newid() UNIQUE NOT NULL,                    -- Global unique record identifier
  PRIMARY KEY           ( agt_ag_id, agt_term_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE acctpl

```
(
  at_partition          INTEGER REFERENCES partition(part_number)
                            NOT FOR REPLICATION,                          -- partition ID to which this template belongs
  at_public             SMALLINT,                                       -- 1 = template is public, 0 = template is private
  at_id                 INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,    -- unique database ID
  at_name               CHAR(32) NOT NULL,                              -- template name
  at_disabled           SMALLINT,                                       -- 1 = badge is disabled, 0 = badge is enabled
  at_event_priv         SMALLINT,                                       -- event priviledge of badge
  at_exec_priv          SMALLINT,                                       -- 1 = executive privilege, 0 = no executive privilege
  at_trace              SMALLINT,                                       -- 1 = badge trace, 0 = no trace
  at_override           SMALLINT,                                       -- 1 = cardholder override enabled, 0 = override disabled
  at_access_grp_0       INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group 1
  at_access_grp_1       INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group 2
  at_access_grp_2       INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group 3
```

```
at_access_grp_3         INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group 4
at_access_grp_4         INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group 5
at_access_grp_5         INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group 6
at_access_grp_6         INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group 7
at_access_grp_7         INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group 8
at_access_tz_0          INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone 1
at_access_tz_1          INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone 2
at_access_tz_2          INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone 3
at_access_tz_3          INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone 4
at_access_tz_4          INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone 5
at_access_tz_5          INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone 6
at_access_tz_6          INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone 7
at_access_tz_7          INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone 8
at_facility_code        INTEGER,                                                       -- badge facility code for Cotag panel
at_priority             SMALLINT,                                                      -- badge priority level
at_special_access_a     SMALLINT,                                                      -- special access flag a
at_special_access_b     SMALLINT,                                                      -- special access flag b
at_special_access_c     SMALLINT,                                                      -- special access flag c
at_security_level       SMALLINT,                                                      -- security level
at_interlock1           INTEGER,                                                       -- bacnet interlock name1
at_interlock1_value     FLOAT,                                                         -- bacnet interlock value1
at_interlock2           INTEGER,                                                       -- bacnet interlock name2
at_interlock2_value     FLOAT,                                                         -- bacnet interlock value2
site                    CHAR(32) NOT NULL,                                             -- Site Name
at_guid                 uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                   -- Global unique record identifier
at_checksum             INTEGER NOT NULL,                                              -- Record checksum
at_access_grp_8         INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 9
at_access_grp_9         INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 10
at_access_grp_10        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 11
at_access_grp_11        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 12
at_access_grp_12        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 13
at_access_grp_13        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 14
at_access_grp_14        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 15
at_access_grp_15        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 16
at_access_grp_16        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 17
at_access_grp_17        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 18
at_access_grp_18        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 19
at_access_grp_19        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 20
at_access_grp_20        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 21
at_access_grp_21        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 22
at_access_grp_22        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 23
at_access_grp_23        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 24
at_access_grp_24        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 25
at_access_grp_25        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 26
at_access_grp_26        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 27
at_access_grp_27        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 28
at_access_grp_28        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 29
at_access_grp_29        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 30
at_access_grp_30        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 31
at_access_grp_31        INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 32
at_access_tz_8          INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 9
at_access_tz_9          INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 10
at_access_tz_10         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 11
at_access_tz_11         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 12
at_access_tz_12         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 13
at_access_tz_13         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 14
at_access_tz_14         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 15
at_access_tz_15         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 16
at_access_tz_16         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 17
at_access_tz_17         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 18
at_access_tz_18         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 19
at_access_tz_19         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 20
at_access_tz_20         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 21
at_access_tz_21         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 22
at_access_tz_22         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 23
```

```
    at_access_tz_23         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 24
    at_access_tz_24         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 25
    at_access_tz_25         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 26
    at_access_tz_26         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 27
    at_access_tz_27         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 28
    at_access_tz_28         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 29
    at_access_tz_29         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 30
    at_access_tz_30         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 31
    at_access_tz_31         INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 32
    PRIMARY KEY             (at_partition, at_id )
);
    CREATE INDEX                at_public ON acctpl ( at_public );
    CREATE UNIQUE INDEX         at_name_part ON acctpl (site, at_name, at_partition );
    CREATE UNIQUE INDEX         at_site_name ON acctpl (site, at_name);
```

CREATE TABLE acctpl_save

```
(
    at_partition            INTEGER NOT NULL,               -- partition ID to which this template belongs
    at_public               SMALLINT,                       -- 1 = template is public, 0 = template is private
    at_id                   INTEGER NOT NULL,               -- unique database ID
    at_name                 CHAR(32) NOT NULL,              -- template name
    at_disabled             SMALLINT,                       -- 1 = badge is disabled, 0 = badge is enabled
    at_event_priv           SMALLINT,                       -- event priviledge of badge
    at_exec_priv            SMALLINT,                       -- 1 = executive privilege, 0 = no executive privilege
    at_trace                SMALLINT,                       -- 1 = badge trace, 0 = no trace
    at_override             SMALLINT,                       -- 1 = cardholder override enabled, 0 = override disabled
    at_access_grp_0         INTEGER NULL,                   -- access group 1
    at_access_grp_1         INTEGER NULL,                   -- access group 2
    at_access_grp_2         INTEGER NULL,                   -- access group 3
    at_access_grp_3         INTEGER NULL,                   -- access group 4
    at_access_grp_4         INTEGER NULL,                   -- access group 5
    at_access_grp_5         INTEGER NULL,                   -- access group 6
    at_access_grp_6         INTEGER NULL,                   -- access group 7
    at_access_grp_7         INTEGER NULL,                   -- access group 8
    at_access_tz_0          INTEGER NULL,                   -- access timezone 1
    at_access_tz_1          INTEGER NULL,                   -- access timezone 2
    at_access_tz_2          INTEGER NULL,                   -- access timezone 3
    at_access_tz_3          INTEGER NULL,                   -- access timezone 4
    at_access_tz_4          INTEGER NULL,                   -- access timezone 5
    at_access_tz_5          INTEGER NULL,                   -- access timezone 6
    at_access_tz_6          INTEGER NULL,                   -- access timezone 7
    at_access_tz_7          INTEGER NULL,                   -- access timezone 8
    at_facility_code        INTEGER,                        -- badge facility code for Cotag panel
    at_priority             SMALLINT,                       -- badge priority level
    at_special_access_a     SMALLINT,                       -- special access flag a
    at_special_access_b     SMALLINT,                       -- special access flag b
    at_special_access_c     SMALLINT,                       -- special access flag c
    at_security_level       SMALLINT,                       -- security level
    at_interlock1           INTEGER,                        -- bacnet interlock name1
    at_interlock1_value     FLOAT,                          -- bacnet interlock value1
    at_interlock2           INTEGER,                        -- bacnet interlock name2
    at_interlock2_value     FLOAT,                          -- bacnet interlock value2
    site                    CHAR(32) NOT NULL,              -- Site Name
    at_guid                 uniqueidentifier NOT NULL,      -- Global unique record identifier
    at_checksum             INTEGER NOT NULL,               -- Record checksum
    upd_guid                uniqueidentifier NOT NULL,      -- Guid for update operation
    upd_mode                SMALLINT NOT NULL,              -- Update mode (0 - delete, 1 - edit, 2 - new)
    upd_timestamp           DATETIME NOT NULL,              -- Date / Time of the update
    upd_checksum            INTEGER NOT NULL,               -- Checksum
    upd_rowguid             uniqueidentifier ROWGUIDCOL DEFAULT
                                newid() UNIQUE NOT NULL,    -- Global unique record identifier
    at_access_grp_8         INTEGER NULL,                   -- access group # 9
    at_access_grp_9         INTEGER NULL,                   -- access group # 10
```

```
  at_access_grp_10      INTEGER NULL,                          -- access group # 11
  at_access_grp_11      INTEGER NULL,                          -- access group # 12
  at_access_grp_12      INTEGER NULL,                          -- access group # 13
  at_access_grp_13      INTEGER NULL,                          -- access group # 14
  at_access_grp_14      INTEGER NULL,                          -- access group # 15
  at_access_grp_15      INTEGER NULL,                          -- access group # 16
  at_access_grp_16      INTEGER NULL,                          -- access group # 17
  at_access_grp_17      INTEGER NULL,                          -- access group # 18
  at_access_grp_18      INTEGER NULL,                          -- access group # 19
  at_access_grp_19      INTEGER NULL,                          -- access group # 20
  at_access_grp_20      INTEGER NULL,                          -- access group # 21
  at_access_grp_21      INTEGER NULL,                          -- access group # 22
  at_access_grp_22      INTEGER NULL,                          -- access group # 23
  at_access_grp_23      INTEGER NULL,                          -- access group # 24
  at_access_grp_24      INTEGER NULL,                          -- access group # 25
  at_access_grp_25      INTEGER NULL,                          -- access group # 26
  at_access_grp_26      INTEGER NULL,                          -- access group # 27
  at_access_grp_27      INTEGER NULL,                          -- access group # 28
  at_access_grp_28      INTEGER NULL,                          -- access group # 29
  at_access_grp_29      INTEGER NULL,                          -- access group # 30
  at_access_grp_30      INTEGER NULL,                          -- access group # 31
  at_access_grp_31      INTEGER NULL,                          -- access group # 32
  at_access_tz_8        INTEGER NULL,                          -- access timezone # 9
  at_access_tz_9        INTEGER NULL,                          -- access timezone # 10
  at_access_tz_10       INTEGER NULL,                          -- access timezone # 11
  at_access_tz_11       INTEGER NULL,                          -- access timezone # 12
  at_access_tz_12       INTEGER NULL,                          -- access timezone # 13
  at_access_tz_13       INTEGER NULL,                          -- access timezone # 14
  at_access_tz_14       INTEGER NULL,                          -- access timezone # 15
  at_access_tz_15       INTEGER NULL,                          -- access timezone # 16
  at_access_tz_16       INTEGER NULL,                          -- access timezone # 17
  at_access_tz_17       INTEGER NULL,                          -- access timezone # 18
  at_access_tz_18       INTEGER NULL,                          -- access timezone # 19
  at_access_tz_19       INTEGER NULL,                          -- access timezone # 20
  at_access_tz_20       INTEGER NULL,                          -- access timezone # 21
  at_access_tz_21       INTEGER NULL,                          -- access timezone # 22
  at_access_tz_22       INTEGER NULL,                          -- access timezone # 23
  at_access_tz_23       INTEGER NULL,                          -- access timezone # 24
  at_access_tz_24       INTEGER NULL,                          -- access timezone # 25
  at_access_tz_25       INTEGER NULL,                          -- access timezone # 26
  at_access_tz_26       INTEGER NULL,                          -- access timezone # 27
  at_access_tz_27       INTEGER NULL,                          -- access timezone # 28
  at_access_tz_28       INTEGER NULL,                          -- access timezone # 29
  at_access_tz_29       INTEGER NULL,                          -- access timezone # 30
  at_access_tz_30       INTEGER NULL,                          -- access timezone # 31
  at_access_tz_31       INTEGER NULL,                          -- access timezone # 32
  PRIMARY KEY           (at_partition, at_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE actioninterlock

```
(
  ai_id                 INTEGER IDENTITY NOT FOR
                            REPLICATION UNIQUE,                -- unique database ID
  ai_partition          INTEGER REFERENCES partition(part_number)
                            NOT FOR REPLICATION,               -- partition ID to which this alarm action interlock belongs
  ai_public             SMALLINT,                              -- 1 = action interlock is public, 0 = action interlock is private
  ai_name               CHAR(32) NOT NULL,                     -- action interlock name
  ai_object_name        CHAR(32) NOT NULL,                     -- BACnet object name this action interlock will modify
  ai_object_prop        SMALLINT NOT NULL,                     -- BACnet object attribute this action interlock will modify
  ai_prop_type          SMALLINT NOT NULL,                     -- 1 = Bool, 3 = Long, 4 = Float, 5 = Double, 16 = Byte, 17 = Short
  ai_priority           SMALLINT NOT NULL,                     -- The BACnet priority to use when writing to the attribute
  site                  CHAR(32) NOT NULL,                     -- Site Name
  ai_guid               uniqueidentifier ROWGUIDCOL UNIQUE
                            NOT NULL,                          -- Global unique record identifier
```

```
  ai_checksum           INTEGER NOT NULL,                          -- Record checksum
  PRIMARY KEY           (ai_partition, ai_id)
);
  CREATE INDEX          ai_site_name ON actioninterlock (site, ai_object_name);
```

CREATE TABLE actioninterlock_save

```
(
  ai_id                 INTEGER NOT NULL,                          -- unique database ID
  ai_partition          INTEGER NOT NULL,                          -- partition ID to which this alarm action interlock belongs
  ai_public             SMALLINT,                                  -- 1 = action interlock is public, 0 = action interlock is private
  ai_name               CHAR(32) NOT NULL,                         -- action interlock name
  ai_object_name        CHAR(32) NOT NULL,                         -- BACnet object name this action interlock will modify
  ai_object_prop        SMALLINT NOT NULL,                         -- BACnet object attribute this action interlock will modify
  ai_prop_type          SMALLINT NOT NULL,                         -- 1 = Bool, 3 = Long, 4 = Float, 5 = Double, 16 = Byte, 17 = Short
  ai_priority           SMALLINT NOT NULL,                         -- The BACnet priority to use when writing to the attribute
  site                  CHAR(32) NOT NULL,                         -- Site Name
  ai_guid               uniqueidentifier NOT NULL,                 -- Global unique record identifier
  ai_checksum           INTEGER NOT NULL,                          -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                 -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                         -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                          -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                            newid() UNIQUE NOT NULL,               -- Global unique record identifier
  PRIMARY KEY           (ai_partition, ai_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE alarmcategory

```
(
  ac_alarmcategory_id   int IDENTITY NOT FOR REPLICATION UNIQUE NOT NULL,
  ac_alarmcategory_url  varchar(256) NOT NULL,
  ac_checksum           int NOT NULL,
  ac_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,
  ac_default            smallint NOT NULL,
  PRIMARY KEY           ( ac_alarmcategory_id )
);
  CREATE UNIQUE INDEX       ac_alarmcategory_url ON alarmcategory (ac_alarmcategory_url);
```

CREATE TABLE alarmcategory_save

```
(
  ac_alarmcategory_id   int NOT NULL,
  ac_alarmcategory_url  varchar (256) NOT NULL,
  ac_checksum           int NOT NULL,
  ac_guid               uniqueidentifier NOT NULL,
  ac_default            smallint NOT NULL,
  upd_guid              uniqueidentifier NOT NULL,
  upd_mode              smallint NOT NULL,
  upd_timestamp         datetime NOT NULL,
  upd_checksum          int NOT NULL,
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,
  PRIMARY KEY           ( ac_alarmcategory_id, upd_guid )
);
```

CREATE TABLE alarmcolors

```
(
  ac_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
  ac_priority           INTEGER NOT NULL,                              -- 0 thru 255
  ac_text_color_pending INTEGER NOT NULL,                              -- Color
  ac_text_color_acked   INTEGER NOT NULL,                              -- Color
  ac_text_color_respond INTEGER NOT NULL,                              -- Color
```

```
  ac_text_color_complete      INTEGER NOT NULL,                    -- Color
  ac_bkgnd_color_pending      INTEGER NOT NULL,                    -- Color
  ac_bkgnd_color_acked        INTEGER NOT NULL,                    -- Color
  ac_bkgnd_color_respond      INTEGER NOT NULL,                    -- Color
  ac_bkgnd_color_complete     INTEGER NOT NULL,                    -- Color
  site                        CHAR(32) NOT NULL,                   -- Site Name
  ac_checksum                 INTEGER NOT NULL,                    -- Record checksum
  PRIMARY KEY                 (ac_priority, site)
);
```

CREATE TABLE alarmcolors_save

```
(
  ac_guid                     uniqueidentifier NOT NULL,           -- Global unique record identifier
  ac_priority                 INTEGER NOT NULL,                    -- 0 thru 255
  ac_text_color_pending       INTEGER NOT NULL,                    -- Color
  ac_text_color_acked         INTEGER NOT NULL,                    -- Color
  ac_text_color_respond       INTEGER NOT NULL,                    -- Color
  ac_text_color_complete      INTEGER NOT NULL,                    -- Color
  ac_bkgnd_color_pending      INTEGER NOT NULL,                    -- Color
  ac_bkgnd_color_acked        INTEGER NOT NULL,                    -- Color
  ac_bkgnd_color_respond      INTEGER NOT NULL,                    -- Color
  ac_bkgnd_color_complete     INTEGER NOT NULL,                    -- Color
  site                        CHAR(32) NOT NULL,                   -- Site Name
  ac_checksum                 INTEGER NOT NULL,                    -- Record checksum
  upd_guid                    uniqueidentifier NOT NULL,           -- Guid for update operation
  upd_mode                    SMALLINT NOT NULL,                   -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp               DATETIME NOT NULL,                   -- Date / Time of the update
  upd_checksum                INTEGER NOT NULL,                    -- Checksum
  upd_rowguid                 uniqueidentifier ROWGUIDCOL DEFAULT
                                newid() UNIQUE NOT NULL,           -- Global unique record identifier
  PRIMARY KEY                 (ac_priority, site, upd_guid)
);
```

CREATE TABLE alarmfw

```
(
  alarmfw_id                  INTEGER IDENTITY
                                NOT FOR REPLICATION UNIQUE,        -- unique database ID
  alarmfw_from_id             INTEGER REFERENCES station(stn_id)
                                NOT FOR REPLICATION,               -- ID of workstation that is forwarding alarms
  alarmfw_to_id               INTEGER REFERENCES station(stn_id)
                                NOT FOR REPLICATION,               -- ID of workstation that will receive forwarded alarms
  alarmfw_mfg_id_station      INTEGER REFERENCES msgfiltergroup(mfg_id)
                                NOT FOR REPLICATION,               -- ID of station message filter group
  alarmfw_mfg_id_user         INTEGER REFERENCES msgfiltergroup(mfg_id)
                                NOT FOR REPLICATION,               -- ID of user message filter group
  site                        CHAR(32) NOT NULL,                   -- Site Name
  alarmfw_guid                uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  alarmfw_checksum            INTEGER NOT NULL,                    -- Record checksum
  PRIMARY KEY                 (alarmfw_id)
);
```

CREATE TABLE alarmfw_save

```
(
  alarmfw_id                  INTEGER NOT NULL,                    -- unique database ID
  alarmfw_from_id             INTEGER NOT NULL,                    -- ID of workstation that is forwarding alarms
  alarmfw_to_id               INTEGER NOT NULL,                    -- ID of workstation that will receive forwarded alarms
  alarmfw_mfg_id_station      INTEGER,                             -- ID of station message filter group
  alarmfw_mfg_id_user         INTEGER,                             -- ID of user message filter group
  site                        CHAR(32) NOT NULL,                   -- Site Name
  alarmfw_guid                uniqueidentifier NOT NULL,           -- Global unique record identifier
  alarmfw_checksum            INTEGER NOT NULL,                    -- Record checksum
```

```
  upd_guid              uniqueidentifier NOT NULL,                    -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                            -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                            -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                             -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,                    -- Global unique record identifier
  PRIMARY KEY           (alarmfw_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE AlarmOptions

```
(
  alo_alarm_type              INTEGER NOT NULL,                       -- Unique alarm type
  alo_alarm_type_guid         uniqueidentifier NOT NULL,              -- Unique ID of each instance of this alarm for this alarm type
  alo_priority                SMALLINT NOT NULL,                      -- priority
  alo_disable_alarm           SMALLINT NOT NULL,                      -- 1 = disable alarm
  alo_popup_alarm             SMALLINT NOT NULL,                      -- 1 = auto popup of alarm monitor for alarm state
  alo_popup_normal            SMALLINT NOT NULL,                      -- 1 = auto popup of alarm monitor for normal state
  alo_popup_others            SMALLINT NOT NULL,                      -- 1 = auto popup of alarm monitor for other states
  alo_inst_alarm              INTEGER NULL REFERENCES alrminst(ai_id)
                                NOT FOR REPLICATION,                  -- instruction text ID for Alarm state
  alo_inst_normal             INTEGER NULL REFERENCES alrminst(ai_id)
                                NOT FOR REPLICATION,                  -- instruction text ID for Normal state
  alo_inst_others             INTEGER NULL REFERENCES alrminst(ai_id)
                                NOT FOR REPLICATION,                  -- instruction text ID for Other states
  alo_event_1                 INTEGER NULL REFERENCES event(ev_id) NOT FOR REPLICATION,
  alo_event_2                 INTEGER NULL REFERENCES event(ev_id) NOT FOR REPLICATION,
  alo_event_3                 INTEGER NULL REFERENCES event(ev_id) NOT FOR REPLICATION,
  alo_event_4                 INTEGER NULL REFERENCES event(ev_id) NOT FOR REPLICATION,
  alo_event_label_1           CHAR(32) NULL,
  alo_event_label_2           CHAR(32) NULL,
  alo_event_label_3           CHAR(32) NULL,
  alo_event_label_4           CHAR(32) NULL,
  alo_alarm_tz                INTEGER NULL REFERENCES timezone(tz_id)
                                NOT FOR REPLICATION,                  -- timezone ID when input
  alo_guid                    uniqueidentifier ROWGUIDCOL
                                UNIQUE NOT NULL,                      -- Global unique record identifier
  alo_checksum                INTEGER NOT NULL,                       -- Record checksum
  alo_category                VARCHAR(256) NOT NULL,
  alo_ack_required            SMALLINT NOT NULL,
  alo_resp_required           SMALLINT NOT NULL,
  alo_associated_av_channel   INTEGER NULL REFERENCES avchannel(cch_id) NOT FOR REPLICATION,
  alo_associated_realtime_map INTEGER NULL REFERENCES map(map_id) NOT FOR REPLICATION,
  alo_escalation_category     SMALLINT NOT NULL,
  alo_escalation_repeat       SMALLINT NOT NULL,
  alo_escalation_timeout      INTEGER NOT NULL,                       --value (1 to 1440)
  alo_escalation_increment    SMALLINT NOT NULL,                      -- value (1 to 10)
  alo_escalation_enable       SMALLINT NOT NULL,
  alo_msea_graphic            uniqueidentifier NULL REFERENCES mseagraphic(mg_guid) NOT FOR REPLICATION,
  PRIMARY KEY                 ( alo_category, alo_alarm_type, alo_alarm_type_guid )
);
  CREATE INDEX                alo_alarm_type_guid ON AlarmOptions ( alo_alarm_type_guid );
```

CREATE TABLE AlarmOptions_save

```
(
  alo_alarm_type              INTEGER NOT NULL,                       -- Unique alarm type
  alo_alarm_type_guid         uniqueidentifier NOT NULL,              -- Unique ID of each instance of this alarm for this alarm type
  alo_priority                SMALLINT,                               -- priority
  alo_disable_alarm           SMALLINT NOT NULL,                      -- 1 = display in alarm monitor
  alo_popup_alarm             SMALLINT NOT NULL,
  alo_popup_normal            SMALLINT NOT NULL,
  alo_popup_others            SMALLINT NOT NULL,
  alo_inst_alarm              INTEGER NULL,                           -- instruction text ID for Alarm state
```

```
  alo_inst_normal              INTEGER NULL,                                  -- instruction text ID for Normal state
  alo_inst_others              INTEGER NULL,                                  -- instruction text ID for Other states
  alo_event_1                  INTEGER NULL,
  alo_event_2                  INTEGER NULL,
  alo_event_3                  INTEGER NULL,
  alo_event_4                  INTEGER NULL,
  alo_event_label_1            CHAR(32) NULL,
  alo_event_label_2            CHAR(32) NULL,
  alo_event_label_3            CHAR(32) NULL,
  alo_event_label_4            CHAR(32) NULL,
  alo_alarm_tz                 INTEGER NULL,                                  -- timezone ID when input
  alo_guid                     uniqueidentifier NOT NULL,
  alo_checksum                 INTEGER NOT NULL,                              -- Record checksum
  upd_guid                     uniqueidentifier NOT NULL,                     -- Guid for update operation
  upd_mode                     SMALLINT NOT NULL,                             -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                DATETIME NOT NULL,                             -- Date / Time of the update
  upd_checksum                 INTEGER NOT NULL,                              -- Checksum
  upd_rowguid                  uniqueidentifier ROWGUIDCOL DEFAULT
                                  newid() UNIQUE NOT NULL,                    -- Global unique record identifier
  alo_category                 VARCHAR(256) NOT NULL,
  alo_ack_required             SMALLINT NOT NULL,
  alo_resp_required            SMALLINT NOT NULL,
  alo_associated_av_channel    INTEGER NULL,
  alo_associated_realtime_map  INTEGER NULL,
  alo_escalation_category      SMALLINT NOT NULL,
  alo_escalation_repeat        SMALLINT NOT NULL,
  alo_escalation_timeout       INTEGER NOT NULL,                              --value (1 to 1440)
  alo_escalation_increment     SMALLINT NOT NULL,                             -- value (1 to 10)
  alo_escalation_enable        SMALLINT NOT NULL,
  alo_msea_graphic             uniqueidentifier NULL,
  PRIMARY KEY                  ( alo_category, alo_alarm_type, alo_alarm_type_guid, upd_guid )
);
```

CREATE TABLE alarmresptext

```
(
  rt_partition          INTEGER REFERENCES partition(part_number)
                           NOT FOR REPLICATION,                  -- partition ID to which this alarm response text belongs
  rt_public             SMALLINT,                                -- 1 = access group is public, 0 = access group is private
  rt_id                 INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,  -- unique database ID
  rt_name               CHAR(32) NOT NULL,                       -- response text name
  rt_text               CHAR(255) NOT NULL,                      -- response text
  site                  CHAR(32) NOT NULL,                       -- Site Name
  rt_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  rt_checksum           INTEGER NOT NULL,                        -- Record checksum
  PRIMARY KEY           ( rt_partition, rt_id )
);
  CREATE INDEX          rt_site_name ON alarmresptext (site, rt_name);
```

CREATE TABLE alarmresptext_save

```
(
  rt_partition          INTEGER NOT NULL,                        -- partition ID to which this alarm response text belongs
  rt_public             SMALLINT,                                -- 1 = access group is public, 0 = access group is private
  rt_id                 INTEGER,                                 -- unique database ID
  rt_name               CHAR(32) NOT NULL,                       -- response text name
  rt_text               CHAR(255) NOT NULL,                      -- response text
  site                  CHAR(32) NOT NULL,                       -- Site Name
  rt_guid               uniqueidentifier UNIQUE NOT NULL,        -- Global unique record identifier
  rt_checksum           INTEGER NOT NULL,                        -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,               -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                       -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                       -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                        -- Checksum
```

```
  upd_rowguid            uniqueidentifier ROWGUIDCOL DEFAULT
                             newid() UNIQUE NOT NULL,                -- Global unique record identifier
  PRIMARY KEY            ( rt_partition, rt_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE alarms

```
(
  al_alarm_id                 INTEGER IDENTITY NOT FOR
                                 REPLICATION PRIMARY KEY,    -- Unique database ID of this alarm
  al_alarm_type               INTEGER NOT NULL,              -- Unique alarm type
  al_alarm_type_id            INTEGER NOT NULL,              -- Unique ID of each instance of this alarm for this alarm type
  al_ack_required             SMALLINT NOT NULL,             -- This alarm needs acked by an operator before it can be completed
  al_instruction_text         VARCHAR(512) NULL,             -- Optional, Instruction text not used unless m_dwInstructionTextID = 0
  al_alarm_state              SMALLINT NOT NULL,             -- Completed = 1, Responding = 2, Acked = 3, Pending = 4, Unknown = 5
  al_state_time               DATETIME NOT NULL,             -- Time this alarm entered the current AlarmState
  al_severity                 SMALLINT NOT NULL,             -- Severity (1=highest) of the alarm
  al_condition_state          SMALLINT NOT NULL,             -- See Alarm Condition States (-1 = N/A)
  al_condition_completion_state SMALLINT NOT NULL,           -- See Alarm Condition States (-1 = N/A)
  al_condition_time           DATETIME NOT NULL,             -- Time of the original alarm
  al_filter_string            VARCHAR(512) NULL,             -- BACnet filter string
  al_popup                    SMALLINT NOT NULL,             -- Should this alarm pop to the front
  al_operator_name            VARCHAR(512) NULL,             -- Last operator to change this alarm state only used if m_ulOperatorID = 0
  al_description              VARCHAR(1024) NOT NULL,        -- To show on report
  al_extra                    IMAGE NULL,                    -- Binary data specific to certain alarm types
  site                        CHAR(32) NULL,                 -- Site name of this alarm originated
  al_operator_site_name       CHAR(32) NULL,                 -- Site name of this alarm acknowleged
  al_partition                CHAR(32) NULL,                 -- Partition name of this alarm
  al_public                   SMALLINT NOT NULL,             -- 0 = private, 1 = public
  al_guid                     uniqueidentifier DEFAULT newid()
                                 ROWGUIDCOL UNIQUE NOT NULL,
  al_condition_seqno          INTEGER NOT NULL,              -- Condition sequence number
  al_checksum                 INTEGER NOT NULL,              -- Record Checksum
  al_category                 CHAR(255) NOT NULL,            -- the Alarm Category for this alarm
  al_escalation_level         SMALLINT NOT NULL,             -- Escalation Level (0 - 10)
  al_alarm_type_guid          uniqueidentifier NULL,         -- Unique GUID of each instance of this alarm for this alarm type
  al_resp_required            SMALLINT NOT NULL,             -- This alarm needs responded by an operator before it can be completed
  al_alarm_options_guid       uniqueidentifier NULL,         -- Unique GUID of the alarm options for this alarm
);
  CREATE INDEX                severity_state_time ON alarms ( al_condition_time, al_severity );
  CREATE INDEX                state_flag ON alarms ( al_alarm_state );
  CREATE INDEX                alarm_site ON alarms ( site );
  CREATE INDEX                alarm_partition ON alarms ( al_partition );
  CREATE INDEX                alarm_public ON alarms ( al_public );
  CREATE INDEX                alarm_guid ON alarms ( al_guid );
```

CREATE TABLE alarms_hist

```
(
  ah_alarm_hist_id            INTEGER IDENTITY NOT FOR
                                 REPLICATION PRIMARY KEY,    -- unique ID of this alarm hist record
  ah_alarm_id                 INTEGER NOT NULL,              -- alarm ID
  ah_alarm_type               INTEGER NOT NULL,              -- Unique alarm type
  ah_alarm_type_id            INTEGER NOT NULL,              -- Unique instance id for this alarm type
  ah_severity                 SMALLINT NOT NULL,             -- Severity (1=highest) of the alarm
  ah_condition_state          SMALLINT NOT NULL,             -- See Alarm Condition States (-1 = N/A)
  ah_condition_completion_state SMALLINT NOT NULL,           -- See Alarm Condition States (-1 = N/A)
  ah_condition_timestamp      DATETIME NOT NULL,             -- Time of the original alarm
  ah_description_string       VARCHAR(1024) NOT NULL,        -- Description of this alarm state
  ah_state_time               DATETIME NOT NULL,             -- Time this alarm entered the current AlarmState
  ah_alarm_state              SMALLINT NOT NULL,             -- Completed = 1, Responding = 2, Acked = 3, Pending = 4, Unknown = 5
  ah_operator_name            VARCHAR(512) NULL,             -- Last operator to change this alarm state (if applicable)
  site                        CHAR(32) NULL,                 -- site name
  ah_operator_site_name       CHAR(32) NULL,                 -- site name of the operator who responded this alarm
```

```
  ah_partition              CHAR(32) NULL,                  -- partition name
  ah_public                 SMALLINT NOT NULL,              -- 0 = private, 1 = public
  ah_alarm_guid             uniqueidentifier,               -- alarm uniqueidentifier
  ah_condition_seqno        INTEGER NOT NULL,               -- Condition sequence number
  ah_guid                   uniqueidentifier ROWGUIDCOL
                              UNIQUE NOT NULL,              -- Global unique record identifier
  ah_checksum               INTEGER NOT NULL,               -- Record checksum
  ah_category               CHAR(255) NOT NULL,             -- the Alarm Category for this alarm
  ah_escalation_level       SMALLINT NOT NULL,              -- Escalation Level (0 - 10)
  ah_alarm_type_guid        uniqueidentifier NULL,          -- Unique GUID of each instance of this alarm for this alarm type
);
  CREATE INDEX              alarm_id ON alarms_hist ( ah_alarm_id );
  CREATE INDEX              alarm_timestamp ON alarms_hist ( ah_condition_timestamp );
  CREATE INDEX              alarm_guid ON alarms_hist ( ah_alarm_guid );
  CREATE INDEX              alarm_site ON alarms_hist ( site );
  CREATE INDEX              alarm_public ON alarms_hist ( ah_public );
  CREATE INDEX              alarm_partition ON alarms_hist ( ah_partition );
  CREATE INDEX              alarm_type ON alarms_hist ( ah_alarm_type );
  CREATE INDEX              alarm_type_id ON alarms_hist ( ah_alarm_type_id );
```

CREATE TABLE alrminst

```
(
  ai_partition              INTEGER REFERENCES partition(part_number)
                              NOT FOR REPLICATION,          -- partition ID to which this alarm instruction belongs
  ai_public                 SMALLINT,                       -- 1 = access group is public, 0 = access group is private
  ai_id                     INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,  -- unique database ID
  ai_name                   CHAR(32) NOT NULL,              -- alarm instruction name
  ai_text                   CHAR(255) NOT NULL,             -- alarm instruction text
  site                      CHAR(32) NOT NULL,              -- Site Name
  ai_guid                   uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  ai_checksum               INTEGER NOT NULL,               -- Record checksum
  PRIMARY KEY               (ai_partition, ai_id)
);
  CREATE UNIQUE INDEX       ai_site_name ON alrminst (site, ai_name);
```

CREATE TABLE alrminst_save

```
(
  ai_partition              INTEGER NOT NULL,               -- partition ID to which this alarm instruction belongs
  ai_public                 SMALLINT,                       -- 1 = access group is public, 0 = access group is private
  ai_id                     INTEGER NOT NULL,               -- unique database ID
  ai_name                   CHAR(32)  NOT NULL,             -- alarm instruction name
  ai_text                   CHAR(255) NOT NULL,             -- alarm instruction text
  site                      CHAR(32) NOT NULL,              -- Site Name
  ai_guid                   uniqueidentifier NOT NULL,      -- Global unique record identifier
  ai_checksum               INTEGER NOT NULL,               -- Record checksum
  upd_guid                  uniqueidentifier NOT NULL,      -- Guid for update operation
  upd_mode                  SMALLINT NOT NULL,              -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp             DATETIME NOT NULL,              -- Date / Time of the update
  upd_checksum              INTEGER NOT NULL,               -- Checksum
  upd_rowguid               uniqueidentifier ROWGUIDCOL DEFAULT
                              newid() UNIQUE NOT NULL,      -- Global unique record identifier
  PRIMARY KEY               (ai_partition, ai_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE alrm_resp

```
(
  id                        INTEGER IDENTITY NOT FOR REPLICATION PRIMARY KEY,  -- unique database ID
  alarm_id                  INTEGER,                        -- unique id of alarm
  action_time               DATETIME,                       -- time when input point changed state
  text                      CHAR(255),                      -- response text
  username                  CHAR(33),                       -- user who entered this response
```

```
  timestamp              DATETIME,                                          -- time that the response was made
  site                   CHAR(32) NOT NULL,                                 -- Site Name
  guid                   uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  checksum               INTEGER NOT NULL,                                  -- Record checksum
);
  CREATE INDEX           alrm_resp_timestamp ON alrm_resp ( action_time );
```

CREATE TABLE [area2terminals]

```
(
  [a2t_guid]             uniqueidentifier ROWGUIDCOL  NOT NULL,
  [a2t_area_id]          [int] NOT NULL,
  [a2t_terminal_id]      [int] NOT NULL,
  [a2t_area_name]        [char] (64) NOT NULL,
  [a2t_terminal_name]    [char] (64) NOT NULL
) ON [PRIMARY]
  ALTER TABLE [area2terminals] WITH NOCHECK ADD
                         CONSTRAINT [PK_area2terminals] PRIMARY KEY  CLUSTERED
                         (
                                [a2t_guid]
                         )  ON [PRIMARY]
  ALTER TABLE [area2terminals] ADD
                         CONSTRAINT [DF_area2terminals_a2t_guid] DEFAULT (newid()) FOR [a2t_guid]
```

CREATE TABLE area_configuration

```
(
  ac_partition           INTEGER REFERENCES partition(part_number)
                           NOT FOR REPLICATION,                             -- partition ID to which this area belongs
  ac_public              SMALLINT,                                          -- 1 = area is public, 0 = area is private
  ac_area_id             INTEGER IDENTITY NOT FOR
                           REPLICATION UNIQUE,                              -- unique database ID
  ac_area_name           char(32) NOT NULL,                                 -- area name
  ac_description         char(64),
  ac_type                SMALLINT NOT NULL,                                 -- 0 = Access, 1 = Facility, 2 = Parking
  ac_entry_mode          SMALLINT NOT NULL,                                 -- 0 = Count All, 1 = Count Access, 2 = Count Input
  ac_exit_mode           SMALLINT NOT NULL,                                 -- 0 = Count All, 1 = Count Access, 2 = Count Input
  ac_pre_max             INTEGER,                                           -- Pre Max Allowed
  ac_max                 INTEGER,                                           -- Max Allowed
  ac_min                 INTEGER,                                           -- Min Required
  ac_correction          INTEGER,                                           -- Correction to the previous three number
  ac_popup_on_max_alarm  SMALLINT NOT NULL,                                 -- 0 = do not pop up, 1 = pop up on alarm
                                                                            -- 2 = pop up on secure, 3 = pop up on both
  ac_popup_on_min_alarm  SMALLINT NOT NULL,                                 -- Same for the status alarm
  ac_popup_on_premax_alarm SMALLINT NOT NULL,                               -- Same for the status alarm
  ac_max_alarm_text_set_id INTEGER NULL REFERENCES alrminst(ai_id)
                           NOT FOR REPLICATION,                             -- Max alarm set instruction text ID if not null
  ac_max_alarm_text_sec_id INTEGER NULL REFERENCES alrminst(ai_id)
                           NOT FOR REPLICATION,                             -- Max alarm secure instruction text
  ac_min_alarm_text_set_id INTEGER NULL REFERENCES alrminst(ai_id)
                           NOT FOR REPLICATION,                             -- Same for the max alarm
  ac_min_alarm_text_sec_id INTEGER NULL REFERENCES alrminst(ai_id)
                           NOT FOR REPLICATION,                             -- Etc.
  ac_premax_alarm_text_set_id INTEGER NULL REFERENCES alrminst(ai_id)
                           NOT FOR REPLICATION,                             -- Same for the max alarm
  ac_premax_alarm_text_sec_id INTEGER NULL REFERENCES alrminst(ai_id)
                           NOT FOR REPLICATION,                             -- Etc
  site                   CHAR(32) NOT NULL,                                 -- Site Name
  ac_guid                uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  ac_checksum            INTEGER NOT NULL,                                  -- Record checksum
  PRIMARY KEY            (ac_partition, ac_area_id )
)
  CREATE UNIQUE INDEX    ac_site_name ON area_configuration (site, ac_area_name);
```

CREATE TABLE area_configuration_save

```
(
  ac_partition                INTEGER NOT NULL,                  -- partition ID to which this area belongs
  ac_public                   SMALLINT,                          -- 1 = area is public, 0 = area is private
  ac_area_id                  INTEGER NOT NULL,                  -- unique database ID
  ac_area_name                char(32) NOT NULL,                 -- area name
  ac_description              char(64),
  ac_type                     SMALLINT NOT NULL,                 -- 0 = Access, 1 = Facility, 2 = Parking
  ac_entry_mode               SMALLINT NOT NULL,                 -- 0 = Count All, 1 = Count Access, 2 = Count Input
  ac_exit_mode                SMALLINT NOT NULL,                 -- 0 = Count All, 1 = Count Access, 2 = Count Input
  ac_pre_max                  INTEGER,                           -- Pre Max Allowed
  ac_max                      INTEGER,                           -- Max Allowed
  ac_min                      INTEGER,                           -- Min Required
  ac_correction               INTEGER,                           -- Correction to the previous three number
  ac_popup_on_max_alarm       SMALLINT NOT NULL,                 -- 0 = do not pop up, 1 = pop up on alarm
                                                                 -- 2 = pop up on secure, 3 = pop up on both
  ac_popup_on_min_alarm       SMALLINT NOT NULL,                 -- Same for the status alarm
  ac_popup_on_premax_alarm    SMALLINT NOT NULL,                 -- Same for the status alarm
  ac_max_alarm_text_set_id    INTEGER NULL,                      -- Max alarm set instruction text ID if not null
  ac_max_alarm_text_sec_id    INTEGER NULL,                      -- Max alarm secure instruction text
  ac_min_alarm_text_set_id    INTEGER NULL,                      -- Same for the max alarm
  ac_min_alarm_text_sec_id    INTEGER NULL,                      -- Etc.
  ac_premax_alarm_text_set_id INTEGER NULL,                      -- Same for the max alarm
  ac_premax_alarm_text_sec_id INTEGER NULL,                      -- Etc
  site                        CHAR(32) NOT NULL,                 -- Site Name
  ac_guid                     uniqueidentifier NOT NULL,         -- Global unique record identifier
  ac_checksum                 INTEGER NOT NULL,                  -- Record checksum
  upd_guid                    uniqueidentifier NOT NULL,         -- Guid for update operation
  upd_mode                    SMALLINT NOT NULL,                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp               DATETIME NOT NULL,                 -- Date / Time of the update
  upd_checksum                INTEGER NOT NULL,                  -- Checksum
  upd_rowguid                 uniqueidentifier ROWGUIDCOL DEFAULT
                                 newid() UNIQUE NOT NULL,        -- Global unique record identifier
  PRIMARY KEY                 (ac_partition, ac_area_id, upd_mode desc, upd_guid )
)
```

CREATE TABLE area_details

```
(
  ad_area_id          INTEGER REFERENCES area_configuration (ac_area_id)
                         NOT FOR REPLICATION,
  ad_control_id       INTEGER NOT NULL,                          --the terminal, termgrp, input, inputgrp ID
  ad_type             SMALLINT NOT NULL,                         --0 = Terminal, 1 = TermGrp, 2 = Input, 3 = InputGrp
  ad_mode             SMALLINT NOT NULL,                         --1 = In, 2 = Out
  site                CHAR(32) NOT NULL,                         -- Site Name
  ad_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  ad_checksum         INTEGER NOT NULL,                          -- Record checksum
  PRIMARY KEY         (ad_area_id, ad_control_id, ad_type, ad_mode)
);
  CREATE INDEX        ad_area_id ON area_details (ad_area_id);
```

CREATE TABLE area_details_save

```
(
  ad_area_id          INTEGER NOT NULL,
  ad_control_id       INTEGER NOT NULL,                          --the terminal, termgrp, input, inputgrp ID
  ad_type             SMALLINT NOT NULL,                         --0 = Terminal, 1 = TermGrp, 2 = Input, 3 = InputGrp
  ad_mode             SMALLINT NOT NULL,                         --1 = In, 2 = Out
  site                CHAR(32) NOT NULL,                         -- Site Name
  ad_guid             uniqueidentifier NOT NULL,                 -- Global unique record identifier
  ad_checksum         INTEGER NOT NULL,                          -- Record checksum
  upd_guid            uniqueidentifier NOT NULL,                 -- Guid for update operation
  upd_mode            SMALLINT NOT NULL,                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp       DATETIME NOT NULL,                         -- Date / Time of the update
```

```
  upd_checksum          INTEGER NOT NULL,                                    -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,                           -- Global unique record identifier
  PRIMARY KEY           (ad_area_id, ad_control_id, ad_type, ad_mode, upd_mode desc, upd_guid)
);
```

CREATE TABLE area_layout

```
(
  al_partition          INTEGER REFERENCES partition(part_number)
                          NOT FOR REPLICATION,                               -- partition ID to which this area layout belongs
  al_public             SMALLINT,                                          -- 1 = area layout is public, 0 = area layout is private
  al_id                 INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,       -- unique database ID
  al_name               char(32) NOT NULL,                                 -- area layout name
  al_description        char(64),
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  al_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  al_checksum           INTEGER NOT NULL,                                  -- Record checksum
  PRIMARY KEY           (al_partition, al_id )
)
  CREATE UNIQUE INDEX     al_site_name ON area_layout (site, al_name);
```

CREATE TABLE area_layout_data

```
(
  ald_id                INTEGER REFERENCES area_layout(al_id) NOT FOR REPLICATION,  -- area layout ID
  ald_item_id           INTEGER,                                           -- negtive = FN, LN, Badge, Company, Dept,
                                                                             DateTime; otherwise the UDF id
  ald_name              char(32),                                          -- name of the items
  ald_order             SMALLINT NOT NULL,                                 -- item order
  ald_width             INTEGER,                                           -- width of the column
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  ald_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  ald_checksum          INTEGER NOT NULL,                                  -- Record checksum
  PRIMARY KEY           (ald_id, ald_item_id, ald_name )
)
```

CREATE TABLE area_layout_data_save

```
(
  ald_id                INTEGER,                                           -- area layout ID
  ald_item_id           INTEGER,                                           -- negtive = FN, LN, Badge, Company, Dept, DateTime;
                                                                             otherwise the UDF id
  ald_name              char(32),                                          -- name of the items
  ald_order             SMALLINT NOT NULL,                                 -- item order
  ald_width             INTEGER,                                           -- width of the column
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  ald_guid              uniqueidentifier NOT NULL,                         -- Global unique record identifier
  ald_checksum          INTEGER NOT NULL,                                  -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                         -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                 -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                  -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                 -- Global unique record identifier
  PRIMARY KEY           (ald_id, ald_item_id, ald_name, upd_mode desc, upd_guid )
)
```

CREATE TABLE area_layout_save

```
(
  al_partition          INTEGER NOT NULL,                                  -- partition ID to which this area layout belongs
  al_public             SMALLINT,                                          -- 1 = area layout is public, 0 = area layout is private
  al_id                 INTEGER NOT NULL,                                  -- unique database ID
```

```
  al_name              char(32) NOT NULL,                          -- area layout name
  al_description       char(64),
  site                 CHAR(32) NOT NULL,                          -- Site Name
  al_guid              uniqueidentifier NOT NULL,                  -- Global unique record identifier
  al_checksum          INTEGER NOT NULL,                           -- Record checksum
  upd_guid             uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode             SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp        DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum         INTEGER NOT NULL,                           -- Checksum
  upd_rowguid          uniqueidentifier ROWGUIDCOL DEFAULT
                         newid() UNIQUE NOT NULL,                  -- Global unique record identifier
  PRIMARY KEY          (al_partition, al_id, upd_mode desc, upd_guid )
)
```

CREATE TABLE area_status

```
(
  area_status_area_id  INTEGER REFERENCES area_configuration (ac_area_id)
                         NOT FOR REPLICATION,
  area_status_counter  INTEGER,                                    -- counter for the area
  area_status_status   SMALLINT NOT NULL,                          -- 0 = Normal, 1 = Pre Max Alarmed, 2 = Max Alarmed,
                                                                      3 = Min Alarmed
  area_status_update   DATETIME,                                   -- time the record was last changed.
  area_status_layout   INTEGER REFERENCES area_layout (al_id)
                         NOT FOR REPLICATION,                      -- the layout id
  site                 CHAR(32) NOT NULL,                          -- Site Name
  area_status_guid     uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  area_status_checksum INTEGER NOT NULL,                           -- Record checksum
  PRIMARY KEY          (area_status_area_id)
);
```

CREATE TABLE area_status_data

```
(
  asd_area_id          INTEGER NOT NULL REFERENCES area_configuration
                         (ac_area_id) NOT FOR REPLICATION,
  asd_area_type        SMALLINT NOT NULL,                          -- area type
  asd_badge            VARCHAR(100) NOT NULL REFERENCES badge (b_number_str)
                         NOT FOR REPLICATION,                      -- badge number
  asd_datetime         DATETIME NOT NULL,                          -- time the record was last changed.
  asd_control          INTEGER NULL REFERENCES terminal (tp_term_id)
                         NOT FOR REPLICATION,                      -- the control id
  site                 CHAR(32) NOT NULL,                          -- Site Name
  asd_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  asd_checksum         INTEGER NOT NULL,                           -- Record checksum
  PRIMARY KEY          (asd_area_id, asd_area_type, asd_badge)
);
  CREATE INDEX         asd_area_id ON area_status_data (asd_area_id);
```

CREATE TABLE assaabloy

```
(
  assa_extended_access_flag      INTEGER NOT NULL,
  assa_override_deadbolt_flag    INTEGER NOT NULL,
  assa_wakeup_communication_flag INTEGER NOT NULL,
  assa_badge_format              varchar(4096) NULL,
  site                           CHAR(32) NOT NULL,
  assa_guid                      uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,
  assa_checksum                  INTEGER NOT NULL,
  PRIMARY KEY                    (assa_guid)
)
```

CREATE TABLE assaabloy_authorization

```
(
  [assabsa_badge_number            VARCHAR(100) NOT NULL PRIMARY KEY,
  [assabsa_authorization_0]        [uniqueidentifier] NULL,
  [assabsa_authorization_1]        [uniqueidentifier] NULL,
  [assabsa_authorization_2]        [uniqueidentifier] NULL,
  [assabsa_authorization_3]        [uniqueidentifier] NULL,
  [assabsa_authorization_4]        [uniqueidentifier] NULL,
  [assabsa_authorization_5]        [uniqueidentifier] NULL,
  [assabsa_authorization_6]        [uniqueidentifier] NULL,
  [assabsa_authorization_7]        [uniqueidentifier] NULL,
  [assabsa_authorization_8]        [uniqueidentifier] NULL,
  [assabsa_authorization_9]        [uniqueidentifier] NULL,
  [assabsa_authorization_10]       [uniqueidentifier] NULL,
  [assabsa_authorization_11]       [uniqueidentifier] NULL,
  [assabsa_authorization_12]       [uniqueidentifier] NULL,
  [assabsa_authorization_13]       [uniqueidentifier] NULL,
  [assabsa_authorization_14]       [uniqueidentifier] NULL,
  [assabsa_authorization_15]       [uniqueidentifier] NULL,
  [assabsa_authorization_16]       [uniqueidentifier] NULL,
  [assabsa_authorization_17]       [uniqueidentifier] NULL,
  [assabsa_authorization_18]       [uniqueidentifier] NULL,
  [assabsa_authorization_19]       [uniqueidentifier] NULL,
  [assabsa_authorization_20]       [uniqueidentifier] NULL,
  [assabsa_authorization_21]       [uniqueidentifier] NULL,
  [assabsa_authorization_22]       [uniqueidentifier] NULL,
  [assabsa_authorization_23]       [uniqueidentifier] NULL,
  [assabsa_authorization_24]       [uniqueidentifier] NULL,
  [assabsa_authorization_25]       [uniqueidentifier] NULL,
  [assabsa_authorization_26]       [uniqueidentifier] NULL,
  [assabsa_authorization_27]       [uniqueidentifier] NULL,
  [assabsa_authorization_28]       [uniqueidentifier] NULL,
  [assabsa_authorization_29]       [uniqueidentifier] NULL,
  [assabsa_authorization_30]       [uniqueidentifier] NULL,
  [assabsa_authorization_31]       [uniqueidentifier] NULL,
  [assabsa_authorization_32]       [uniqueidentifier] NULL,
  [assabsa_authorization_33]       [uniqueidentifier] NULL,
  [assabsa_authorization_34]       [uniqueidentifier] NULL,
  [assabsa_authorization_35]       [uniqueidentifier] NULL,
  [assabsa_authorization_36]       [uniqueidentifier] NULL,
  [assabsa_authorization_37]       [uniqueidentifier] NULL,
  [assabsa_authorization_38]       [uniqueidentifier] NULL,
  [assabsa_authorization_39]       [uniqueidentifier] NULL,
  [assabsa_authorization_40]       [uniqueidentifier] NULL,
  [assabsa_authorization_41]       [uniqueidentifier] NULL,
  [assabsa_authorization_42]       [uniqueidentifier] NULL,
  [assabsa_authorization_43]       [uniqueidentifier] NULL,
  [assabsa_authorization_44]       [uniqueidentifier] NULL,
  [assabsa_authorization_45]       [uniqueidentifier] NULL,
  [assabsa_authorization_46]       [uniqueidentifier] NULL,
  [assabsa_authorization_47]       [uniqueidentifier] NULL,
  [assabsa_authorization_48]       [uniqueidentifier] NULL,
  [assabsa_authorization_49]       [uniqueidentifier] NULL,
  [assabsa_authorization_50]       [uniqueidentifier] NULL,
  [assabsa_authorization_51]       [uniqueidentifier] NULL,
  [assabsa_authorization_52]       [uniqueidentifier] NULL,
  [assabsa_authorization_53]       [uniqueidentifier] NULL,
  [assabsa_authorization_54]       [uniqueidentifier] NULL,
  [assabsa_authorization_55]       [uniqueidentifier] NULL,
  [assabsa_authorization_56]       [uniqueidentifier] NULL,
  [assabsa_authorization_57]       [uniqueidentifier] NULL,
  [assabsa_authorization_58]       [uniqueidentifier] NULL,
  [assabsa_authorization_59]       [uniqueidentifier] NULL,
  [assabsa_authorization_60]       [uniqueidentifier] NULL,
```

```
  [assabsa_authorization_61]        [uniqueidentifier] NULL,
  [assabsa_authorization_62]        [uniqueidentifier] NULL,
  [assabsa_authorization_63]        [uniqueidentifier] NULL
)
```

CREATE TABLE assaabloy_badgeformat

```
(
  aabf_name                 VARCHAR(32) NOT NULL,                    -- badge format name
  aabf_technology           INTEGER NOT NULL,                        -- badge technology (0 = not specified, 1 = MagStripe,
                                                                        2 = Prox, 3 = IClass, 4 = SmartCard)
  aabf_bits                 INTEGER NOT NULL,                        -- number of bits in badge format
  aabf_qualifier            INTEGER NOT NULL,                        -- qualifier of badge format
  aabf_badge_format         VARCHAR(4096) NOT NULL,                  -- badge format
  site                      VARCHAR(32) NOT NULL,                    -- Site Name
  aabf_guid                 uniqueidentifier ROWGUIDCOL DEFAULT
                              newid() UNIQUE NOT NULL,               -- Global unique record identifier
  aabf_checksum             INTEGER NOT NULL,                         -- Record checksum
  PRIMARY KEY               (aabf_technology, aabf_bits, aabf_qualifier)
);
```

CREATE TABLE assaabloy_badgeformat_save

```
(
  aabf_name                 VARCHAR(32) NOT NULL,                    -- badge format name
  aabf_technology           INTEGER NOT NULL,                        -- badge technology (0 = not specified, 1 = MagStripe,
                                                                        2 = Prox, 3 = IClass, 4 = SmartCard)
  aabf_bits                 INTEGER NOT NULL,                        -- number of bits in badge format
  aabf_qualifier            INTEGER NOT NULL,                        -- qualifier of badge format
  aabf_badge_format         VARCHAR(4096) NOT NULL,                  -- badge format
  site                      VARCHAR(32) NOT NULL,                    -- Site Name
  aabf_guid                 uniqueidentifier NOT NULL,               -- Global unique record identifier
  aabf_checksum             INTEGER NOT NULL,                        -- Record checksum
  upd_guid                  uniqueidentifier NOT NULL,               -- Guid for update operation
  upd_mode                  SMALLINT NOT NULL,                       -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp             DATETIME NOT NULL,                       -- Date / Time of the update
  upd_checksum              INTEGER NOT NULL,                        -- Checksum
  upd_rowguid               uniqueidentifier ROWGUIDCOL DEFAULT
                              newid() UNIQUE NOT NULL,               -- Global unique record identifier
  PRIMARY KEY               (aabf_technology, aabf_bits, aabf_qualifier, upd_mode desc, upd_guid)
);
```

CREATE TABLE assaabloy_download

```
(
  assad_p2000_id            INTEGER NOT NULL,
  assad_type                SMALLINT NOT NULL,
  assad_dsr_id              uniqueidentifier NOT NULL,
  assad_dsr_parent_id       uniqueidentifier NULL,
  assad_guid                uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,
  PRIMARY KEY               (assad_guid)
)
```

CREATE TABLE IntegrationComponent_assaabloy_dsr

```
(
  [dsr_igc_id]              INTEGER REFERENCES IntegrationComponent(igc_ic_id)
                              NOT FOR REPLICATION NOT NULL,
  [dsr_encryption]          SMALLINT NULL,
  site                      CHAR(32) NOT NULL,
  [dsr_guid]                [uniqueidentifier] ROWGUIDCOL
                              PRIMARY KEY DEFAULT newid() NOT NULL,
  [dsr_checksum]            [int] NOT NULL,
)
```

CREATE TABLE IntegrationComponent_assaabloy_dsr_save

```
(
  dsr_igc_id                INTEGER NOT NULL,
  dsr_encryption            SMALLINT NULL,
  site                      CHAR(32) NOT NULL,
  dsr_guid                  uniqueidentifier NOT NULL,
  dsr_checksum              INTEGER NOT NULL,
  upd_guid                  uniqueidentifier NOT NULL,          -- Guid for update operation
  upd_mode                  SMALLINT NOT NULL,                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp             DATETIME NOT NULL,                  -- Date / Time of the update
  upd_checksum              INTEGER NOT NULL,                   -- Checksum
  upd_rowguid               uniqueidentifier ROWGUIDCOL DEFAULT
                              newid() UNIQUE NOT NULL,          -- Global unique record identifier
  PRIMARY KEY               ( dsr_guid, upd_mode desc, upd_guid )
)
```

CREATE TABLE [dbo].[assaabloy_holiday_dayexception]

```
(
  [assaphde_ph_guid]        [uniqueidentifier] NOT NULL,
  [assaphde_de_guid]        [uniqueidentifier] NOT NULL,
  [assaphde_de_group_guid]  [uniqueidentifier] NULL,
  [site]                    [char](32) NOT NULL,
  primary key               (assaphde_ph_guid, assaphde_de_guid)
)
```

CREATE TABLE assaabloy_save

```
(
  assa_extended_access_flag     INTEGER NOT NULL,
  assa_override_deadbolt_flag   INTEGER NOT NULL,
  assa_wakeup_communication_flag INTEGER NOT NULL,
  assa_badge_format             varchar(4096) NOT NULL,
  site                          CHAR(32) NOT NULL,
  assa_guid                     [uniqueidentifier] NOT NULL,
  assa_checksum                 INTEGER NOT NULL,
  upd_guid                      uniqueidentifier NOT NULL,      -- Guid for update operation
  upd_mode                      SMALLINT NOT NULL,              -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                 DATETIME NOT NULL,              -- Date / Time of the update
  upd_checksum                  INTEGER NOT NULL,               -- Checksum
  upd_rowguid                   uniqueidentifier ROWGUIDCOL DEFAULT
                                  newid() UNIQUE NOT NULL,      -- Global unique record identifier
  PRIMARY KEY                   ( assa_guid, upd_mode desc, upd_guid )
);
```

CREATE TABLE [dbo].[assaabloy_timezone_dayperiod]

```
(
  [assatd_tz_id]            [integer] NOT NULL,
  [assatd_tz_guid]          [uniqueidentifier] primary key NOT NULL,
  [assatd_day_0_dp_guid]    [uniqueidentifier] NULL,
  [assatd_day_1_dp_guid]    [uniqueidentifier] NULL,
  [assatd_day_2_dp_guid]    [uniqueidentifier] NULL,
  [assatd_day_3_dp_guid]    [uniqueidentifier] NULL,
  [assatd_day_4_dp_guid]    [uniqueidentifier] NULL,
  [assatd_day_5_dp_guid]    [uniqueidentifier] NULL,
  [assatd_day_6_dp_guid]    [uniqueidentifier] NULL,
  [assatd_deg_guid]         [uniqueidentifier] NULL,
)
```

CREATE TABLE assaabloy_user

```
(
  [assau_number_str]      VARCHAR(100) REFERENCES badge(b_number_str)
                            NOT FOR REPLICATION NOT NULL,
  [assau_user_guid]       uniqueidentifier NOT NULL,
  [assau_guid]            [uniqueidentifier] ROWGUIDCOL  NOT NULL,
  [assau_checksum]        INTEGER NOT NULL,
  PRIMARY KEY             (assau_number_str)
)
```

CREATE TABLE assaabloy_user_save

```
(
  [assau_number_str]      VARCHAR(100) NOT NULL,
  [assau_user_guid]       [uniqueidentifier] NOT NULL,
  [assau_guid]            [uniqueidentifier] NOT NULL,
  [assau_checksum]        [int] NOT NULL,
  upd_guid                uniqueidentifier NOT NULL,          -- Guid for update operation
  upd_mode                SMALLINT NOT NULL,                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp           DATETIME NOT NULL,                  -- Date / Time of the update
  upd_checksum            INTEGER NOT NULL,                   -- Checksum
  upd_rowguid             uniqueidentifier ROWGUIDCOL DEFAULT
                            newid() UNIQUE NOT NULL,          -- Global unique record identifier
  PRIMARY KEY             ( assau_number_str, upd_mode desc, upd_guid )
)
```

CREATE TABLE audit

```
(
  au_timestamp            DATETIME,                   -- time the audit entry was made
  au_username             CHAR(16) NULL,              -- user who generated the audit message
  au_item_type            SMALLINT,                   -- category, such as changing a cardholder
                            1 = User, 2 = Badge, 3 = Badge layout, 4 = Badge fields,
                            5 = Badge EN code, 6 = ID badge, 7 = Cardholder, 8 = Panel,
                            9 = Terminal, 10 = Partition, 11 = Terminal group,
                            12 = Access group, 13 = Holiday, 14 = Timezone, 15 = Input,
                            16 = Input group, 17 = Panel holiday, 18 = Access template,
                            19 = Alarm response text, 20 = Alarm instruction, 21 = Company,
                            22 = Output, 23 = Output group, 24 = Department,
                            25 = Panel timezone, 26 = Softalarm, 27 = Site, 28 = Station,
                            29 = Map, 30 = Image set, 31 = UDF gen, 32 = Event,
                            33 = Panel card event, 34 = Alarms filter, 35 = Alarms forwarding,
                            36 = <<unused>>, 37 = Permissions group, 38 = Panel relay,
                            39 = Report, 40 = MIS interface, 41 = Image recall filter,
                            42 = Counter, 43 = Action interlock, 44 = BACnet source,
                            45 = Guard Tour definition, 46 = Guard Tour Station,
                            47 = Panel Communication Loop, 48 = Elevator,
                            49 = Elevator Floor Mask, 50 = Elevator Floor Group,
                            51 = Elevator Floor Name, 52 = Cabinet, 53 = Cabinet Door Group,
                            54 = Cabinet Door Mask, 55 = Cabinet Door Name, 56 = Area,
                            57 = Zone, 58 = Area Layout, 59 = Connection, 60 = CCTV Server,
                            61 = CCTV Switch, 62 = CCTV Tour, 63 = CCTV Alarm,
                            64 = CCTV Macro, 65 = CCTV Aux, 66 = CCTV Monitor,
                            67 = CCTV Sequence, 68 = CCTV Camera, 69 = CCTV Preset,
                            70 = CCTV Pattern, 71 = CCTV Aux Camera, 72 = Enable Code,
                            73 = P900 Flag, 74 = P900 Counter, 75 = P900 Event,
                            76 = P900 Link, 77 = P900 Sys Parameter, 78 = Auto Badge,
                            79 = Air Crew Pin, 80 = P900 Seq File, 81 = Remote Server,
                            82 = Message Filter, 83 = Message Filter Group, 84 = Local Site,
                            85 = Service, 86 = Application, 87 = Panel Card Format,
                            88 = Badge Reason, 89 = Security Level Range, 90 = Import File,
                            91 = Import Consolidation, 92 = Import Badge Format,
                            93 = Import TCPIP, 94 = Audit, 95 = Alarm History, 96 = Alarm,
                            97 = Generic Text, 98 = Muster History, 99 = Guard Tour History,
```

```
                                                    100 = Xaction History, 101 = Redundancy,
                                                    102 = Data Import Mapping, 103 = Data Import Fields,
                                                    104 = Intercom Exchange, 105 = Intercom Station, 106 = AV Site,
                                                    107 = AV Channel, 108 = AV Monitor, 109 = AV Preset,
                                                    110 = AV Input to Camera, 111 = <<unused>>,
                                                    112 = Enterprise Site, 113 = Enterprise Parameter,
                                                    114 = AV Dry Contact, 115 = Alarm Colors, 116 = Badge Setup,
                                                    117 = Request Approver, 118 = FASCN CCC, 119 = Badge Purpose
  au_action             SMALLINT,                 -- action taken, such as add
                                                    0 = Execute, 1 = Logon, 2 = Logoff, 3 = Add, 4 = Edit, 5 = Delete,
                                                    6 = Print, 7 = Download, 8 = Set, 9 = Reset, 10 = Lock,
                                                    11 = Unlock, 12 = Timed override, 13 = Lock all, 14 = Unlock all,
                                                    15 = Update, 16 = Write flash, 17 = Start Muster, 18 = Stop Muster,
                                                    19 = Demuster, 20 = <<unused>>, 21 = Zone Ready,
                                                    22 = Start Muster Drill, 23 = <<unused>>, 24 = <<unused>>,
                                                    25 = Print Group, 26 = Remove Badge, 27 = Expand Zone,
                                                    28 = Enable Print, 29 = Disable Print, 30 = Save Data, 31 = Pulse,
                                                    32 = Enable, 33 = Clear, 34 = Disable, 35 = Force, 36 = Calibrate,
                                                    37 = Uncalibrate, 38 = Set In Use, 39 = Set Available,
                                                    40 = Manual Trigger, 41 = Store Default, 42 = Security Level,
                                                    43 = Logon Invalid, 44 = Logon Disabled, 45 = FDA Backup,
                                                    46 = FDA Violation, 47 = FDA Checksum Calc,
                                                    48 = Redundancy Failover, 49 = Redundancy Offline,
                                                    50 = Redundancy Online, 51 = Redundancy Mirror,
                                                    52 = Redundancy Monitor, 53 = Redundancy Admin,
                                                    54 = Redundancy Reboot, 55 = Redundancy Config Export,
                                                    56 = Redundancy Config Import, 57 = Download Firmware,
                                                    58 = Verify Firmware, 59 = Apply Firmware, 60 = Erase Database,
                                                    61 = Reboot, 62 = Resync In, 63 = Resync Out,
                                                    64 = Resync Undefined, 65 = Supporess, 66 = Emergency Disable
  au_item_id            INTEGER NULL,             -- ID of the item modified by the action
  au_item_name          CHAR(32) NULL,            -- name of the item modified by the action
  site                  CHAR(32) NULL,            -- site name
  au_partition_name     CHAR(32) NOT NULL,        -- partition name of item routed from other P2000 sites
  au_public             SMALLINT NOT NULL,        -- 1 = public, 0 = private
  au_guid               uniqueidentifier DEFAULT newid()
                          ROWGUIDCOL UNIQUE NOT NULL,
  au_timestamp_utc      DATETIME NOT NULL,
  au_checksum           INTEGER NOT NULL,         -- Record checksum
  PRIMARY KEY           NONCLUSTERED ( au_partition_name, au_guid )
);
  CREATE CLUSTERED INDEX    au_timestamp ON audit ( au_timestamp );
  CREATE INDEX              au_site ON audit ( site);
  CREATE INDEX              au_partition ON audit ( au_partition_name );
  CREATE INDEX              au_public ON audit ( au_public );
  CREATE INDEX              au_item_type ON audit ( au_item_type );
  CREATE INDEX              au_action ON audit ( au_action );
```

CREATE TABLE autobadge

```
(
  ab_partition          INTEGER NOT NULL REFERENCES partition(part_number)
                          NOT FOR REPLICATION,      -- partition ID to which this badge belongs
  ab_public             SMALLINT NOT NULL,        -- 1 = number is public, 0 = number is private
  ab_number_str         VARCHAR(100) PRIMARY KEY NOT NULL, -- badge number
  ab_type               SMALLINT NOT NULL,        -- 0=Regular 1=Visitor
  ab_issue              SMALLINT NOT NULL,        -- issue level
  ab_status             SMALLINT NOT NULL,        -- current status of this auto badge number 0=Available
                                                    1=Reserved 2=In Use
  ab_modification_time  DATETIME NOT NULL,        -- time of last status change
  site                  CHAR(32) NOT NULL,        -- Site Name
  ab_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  ab_checksum           INTEGER NOT NULL,         -- Record checksum
);
```

```
CREATE INDEX        ab_partition ON autobadge ( ab_partition );
CREATE INDEX        ab_status ON autobadge ( ab_status );
```

CREATE TABLE auto_cardholder_id

```
(
  acid_guid                       uniqueidentifier ROWGUIDCOL
                                    UNIQUE NOT NULL,          -- Global unique record identifier
  acid_min_length                 SMALLINT NOT NULL,          -- for display of leading zeros
  acid_enable                     SMALLINT NOT NULL,          -- 1 = enable 0 = disable
  acid_starting_number            INT NOT NULL,               -- default is 1
  acid_prevent_cardholder_id_edit SMALLINT NOT NULL,          -- 1 = allow user of cardholder edit to change auto id
  site                            CHAR(32) NOT NULL,          -- Site Name
  acid_checksum                   INTEGER NOT NULL,           -- Record checksum
);
```

CREATE TABLE auto_cardholder_id_save

```
(
  acid_guid                       uniqueidentifier NOT NULL,  -- Global unique record identifier
  acid_min_length                 SMALLINT NOT NULL,          -- for display of leading zeros
  acid_enable                     SMALLINT NOT NULL,          -- 1 = enable 0 = disable
  acid_starting_number            INT NOT NULL,               -- default is 1
  acid_prevent_cardholder_id_edit SMALLINT NOT NULL,          -- 1 = allow user of cardholder edit to change auto id
  site                            CHAR(32) NOT NULL,          -- Site Name
  acid_checksum                   INTEGER NOT NULL,           -- Record checksum
  upd_guid                        uniqueidentifier NOT NULL,  -- Guid for update operation
  upd_mode                        SMALLINT NOT NULL,          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                   DATETIME NOT NULL,          -- Date / Time of the update
  upd_checksum                    INTEGER NOT NULL,           -- Checksum
  upd_rowguid                     uniqueidentifier ROWGUIDCOL DEFAULT
                                    newid() UNIQUE NOT NULL,  -- Global unique record identifier
);
```

CREATE TABLE auto_cardholder_status

```
(
  acs_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  acs_last_used_id  INTEGER NULL,                                 -- Last used id. If NULL
  site              CHAR(32) NOT NULL,                            -- Site Name
  acs_checksum      INTEGER NOT NULL,                             -- Record checksum
);
```

CREATE TABLE [AVChannel]

```
(
  [CCH_ID]                [int] IDENTITY NOT FOR REPLICATION NOT NULL,  -- Unique ID of channel record
  [CCH_ParentID]          [int] NOT NULL,                               -- ID of parent record
  [CCH_Name]              [char] (64) NOT NULL,                         -- OPC name of channel
  [CCH_ServerID]          [int] NOT NULL,                               -- ID of server record
  [CCH_Exists]            [int] NOT NULL CONSTRAINT
                            [DF_AVChannel_Exists] DEFAULT (1),          -- 0 = No, 1 = Yes, 2 = Use default
  [CCH_Description]       [char] (64) NOT NULL,                         -- Description
  [CCH_Partition]         [int] NOT NULL CONSTRAINT
                            [DF_AVChannel_Partition] DEFAULT (0),       -- Partition ID to which this camera belongs
  [CCH_Public]            [int] NOT NULL,                               -- 1 = camera is public, 0 = camera is private
  [CCH_GeneralString]     [char] (64) NULL,                             -- Camera annotation
  [CCH_PathDS]            [char] (255) NOT NULL,                        -- certified name of channel path such as
                                                                          dvr.swtich.channel
  [CCH_ClientLockId]      [char] (32) NULL,                             -- client id who locks the camera
  [CCH_ChannelStatus]     [int] NOT NULL DEFAULT (0),                   -- indicates the current status of a channel
-- VCR operation
  [CCH_StartRecordExists] [int] NOT NULL CONSTRAINT
                            [DF_AVChannel_StartRecordExists] DEFAULT (1),  -- 0 = No, 1 = Yes, 2 = Use default
```

```
    [CCH_StartRecord]                   [int] NOT NULL CONSTRAINT [
                                            DF_AVChannel_StartRecord] DEFAULT (0),          -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_StopRecordExists]              [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_StopRecordExits] DEFAULT (0),     -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_StopRecord]                    [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_StopRecord] DEFAULT (0),
    [CCH_RewindExists]                  [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_RewindExists] DEFAULT (0),        -- 0 = No, 1 = yes, 2 = use default
    [CCH_Rewind]                        [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_Rewind] DEFAULT (0),              -- 0 = no, 1 = yes, 2 = use default
    [CCH_PauseExists]                   [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_PauseExists] DEFAULT (0),         -- 0=no, 1=yes, 2=use default
    [CCH_Pause]                         [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_Pause] DEFAULT (0),               -- 0=no, 1=yes, 2=use default
    [CCH_FastForwardExists]             [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_FastForwardExists] DEFAULT (0),   -- 0=no, 1=yes, 2=use default
    [CCH_FastForward]                   [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_FastForward] DEFAULT (0),         -- 0=no, 1=yes, 2=use default
    [CCH_StartPlaybackExists]           [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_StartPalybackExists] DEFAULT (0), -- 0=no, 1=yes, 2=use default
    [CCH_StartPlayback]                 [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_StartPalyback] DEFAULT (0),       -- 0=no, 1=yes, 2=use default
    [CCH_StopPlaybackExists]            [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_StopPalybackExists] DEFAULT (0),  -- 0=no, 1=yes, 2=use default
    [CCH_StopPlayback]                  [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_StopPalyback] DEFAULT (0),        -- 0=no, 1=yes, 2=use default
-- Misc camera
    [CCH_ArchiveExportExists]           [int] not null constraint
                                            [DF_AVChannel_ArchiveExportExists] default (0), -- 0=no, 1= yes, 2= use default
    [CCH_ArchiveExport]                 [int] not null constraint [DF_AVChannel_ArchiveExport] default (0),
    [CCH_SetPreAlarmExists]             [int] not null constraint [DF_AVChannel_PreAlarmExists] default (0),
    [CCH_SetPreAlarm]                   [int] not null constraint [DF_AVChannel_SetPreAlarm] default (0),
    [CCH_SetPostAlarmExists]            [int] not null constraint [DF_AVChannel_SetPostAlarmExists] default (0),
    [CCH_SetPostAlarm]                  [int] not null constraint [DF_AVChannel_SetPostAlarm] default (0),
    [CCH_SetVideoLossAlarmExists]       [int] not null constraint [DF_AVChannel_SetVideoLossAlarmExists] default (0),
    [CCH_SetVideoLossAlarm]             [int] not null constraint [DF_AVChannel_SetVideoLossAlarm] default (0),
    [CCH_SetMotionDetectionAlarmExists] [int] not null constraint [DF_AVChannel_SetMotionDetectionAlarmExists] default (0),
    [CCH_SetMotionDetectionAlarm]       [int] not null constraint [DF_AVChannel_SetMotionDetectionAlarm] default (0),
    [CCH_SetFrameRateExists]            [int] not null constraint [DF_AVChannel_SetFrameRateExists] default (0),
    [CCH_SetFrameRate]                  [int] not null constraint [DF_AVChannel_SetFrameRate] default (0),
    [CCH_SetResolutionExists]           [int] not null constraint [DF_AVChannel_SetResolutionExists] default (0),
    [CCH_SetResolution]                 [int] not null constraint [DF_AVChannel_SetResolution] default (0),
-- Camera positions
    [CCH_PresetExists]                  [int] not null constraint [DF_AVChannel_PresetExists] default (0),
    [CCH_PresetMax]                     [int] not null constraint [DF_AVChannel_PresetMax] default ((-1)),      -- number of presets for this cameara
    [CCH_PresetStopExists]              [int] not null constraint [DF_AVChannel_PresetStopExists] default (0),
    [CCH_PresetStop]                    [int] not null constraint [DF_AVChannel_PresetStop] default (0),
    [CCH_PresetRecordExists]            [int] not null constraint [DF_AVChannel_PresetRecordExists] default (0),
    [CCH_PresetRecord]                  [int] not null constraint [DF_AVChannel_PresetRecord] default (0),
    [CCH_PresetPlayExists]              [int] not null constraint [DF_AVChannel_PresetPlayExists] default (0),
    [CCH_PresetPlay]                    [int] not null constraint [DF_AVChannel_PresetPlay] default (0),
-- PT control
    [CCH_TiltExists]                    [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_TiltExists] DEFAULT (0),          -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_Tilt]                          [int] not null constraint [DF_AVChannel_Tilt] default (0),
    [CCH_PanExists]                     [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_PanExists] DEFAULT (0),           -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_Pan]                           [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_Pan] DEFAULT (0),                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_StopAllPanTilt]                [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_StopAllPanTilt] DEFAULT (0),      -- 0 = No, 1 = Yes, 2 = Use default
-- Zoom and lense control
    [CCH_ZoomExists]                    [int] NOT NULL CONSTRAINT
                                            [DF_AVChannel_ZoomExists] DEFAULT (0),          -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_Zoom]                          [int] NOT NULL CONSTRAINT
```

```
                                    [DF_AVChannel_Zoom] DEFAULT (0),                -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_FocusExists]               [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_FocusExists] DEFAULT (0),         -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_Focus]                     [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_Focus] DEFAULT (0),               -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_IrisExists]                [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_IrisExists] DEFAULT (0),          -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_IrisAutomatic]             [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_IrisAutomatic] DEFAULT (0),       -- 0 = No, 1 = Yes
    [CCH_Iris]                      [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_Iris] DEFAULT (0),                -- negative for close = No, positive for open
    [CCH_LensSpeed]                 [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_LensSpeed] DEFAULT (1),           -- Lens speed
    [CCH_LensSpeedMax]              [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_LensSpeedMax] DEFAULT (1),        -- Maximun lens speed, 1 = speed not variable
    [CCH_StopAllZoomFocusIris]      [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_StopAllZoomFocusIrsi] DEFAULT (0), -- 0 = No, 1 = Yes, 2 = Use default
-- other channel mainipulation command and status
    [CCH_Arm]                       [int] not null constraint [DF_AVChannel_Arm] DEFAULT (0),      -- 0=no, 1 = yes, 2 = use default
    [CCH_DisArm]                    [int] not null constraint [DF_AVChannel_DisArm] default (0),
    [CCH_IsArmed]                   [int] not null constraint [DF_AVChannel_IsArmed] default (0),  -- check arm status
    [CCH_WiperExists]               [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_WiperExists] DEFAULT (0),         -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_Wiper]                     [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_Wiper] DEFAULT (0),               -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_WasherExists]              [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_WasherExists] DEFAULT (0),        -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_Washer]                    [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_Washer] DEFAULT (0),              -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_LightExists]               [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_LightExists] DEFAULT (1),         -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_Light]                     [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_Light] DEFAULT (0),               -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_StatusExists]              [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_StatusExists] DEFAULT (1),        -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_AuxiliaryExists]           [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_AuxiliaryExists] DEFAULT (2),     -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_AuxiliaryCount]            [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_AuxiliaryCount] DEFAULT (0),      -- Number configured in database
    [CCH_AuxiliaryMax]              [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_AuxiliaryMax] DEFAULT ((-1)),     -- Number created in namespace,
                                                                                       -1 = Use default
    [CCH_AuxiliaryPlayExists]       [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_AuxiliaryPlayExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_AuxiliaryStopExists]       [int] NOT NULL CONSTRAINT
                                    [DF_AVChannel_AuxiliaryStopExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
    site                            CHAR(32) NOT NULL,                              -- Site Name
    CCH_guid                        uniqueidentifier ROWGUIDCOL
                                    DEFAULT newid() UNIQUE  NOT NULL,               -- Global unique record identifier
    CCH_checksum                    INTEGER NOT NULL,                               -- Record checksum
    [CCH_alarm_debounce_motion]     SMALLINT NOT NULL,                              -- alarm debounce time
    [CCH_alarm_debounce_behavior]   SMALLINT NOT NULL,                              -- alarm debounce time
    [CCH_ext_id_address]            [int] NOT NULL,
    [CCH_ext_guid_address]          [uniqueidentifier] NULL,
    [CCH_device_ip_address]         [char](16) NULL,
    [CCH_ext_encoder]               [int] NOT NULL,
    [CCH_xml_info]                  nvarchar(1024) NULL,
    CONSTRAINT [PK_AVChannel] PRIMARY KEY  CLUSTERED
    (
                                    [CCH_ID]
    )  ON [PRIMARY],
    CONSTRAINT [FK_AVChannel_AVSite] FOREIGN KEY
    (
                                    [CCH_ParentID]
    ) REFERENCES [AVSite] (
```

```
                                        [CAV_ID]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_AVChannel_CCTVServer] FOREIGN KEY
  (
                                        [CCH_ServerID]
  ) REFERENCES [CCTVServer] (
                                        [CSV_ID]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_AVChannel_partition] FOREIGN KEY
  (
                                        [CCH_Partition]
  ) REFERENCES [partition] (
                                        [part_number]
  ) NOT FOR REPLICATION
) ON [PRIMARY]
;
```

CREATE TABLE [AVChannel_save]

```
(
  [CCH_ID]                              [int] NOT NULL,          -- Unique ID of channel record
  [CCH_ParentID]                        [int] NOT NULL,          -- ID of parent record
  [CCH_Name]                            [char] (64) NOT NULL,    -- OPC name of channel
  [CCH_ServerID]                        [int] NOT NULL,          -- ID of server record
  [CCH_Exists]                          [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCH_Description]                     [char] (64) NOT NULL,    -- Description
  [CCH_Partition]                       [int] NOT NULL,          -- Partition ID to which this camera belongs
  [CCH_Public]                          [int] NOT NULL,          -- 1 = camera is public, 0 = camera is private
  [CCH_GeneralString]                   [char] (64) NULL,        -- Camera annotation
  [CCH_PathDS]                          [char] (255) NOT NULL,   -- certified name of channel path such as dvr.swtich.channel
  [CCH_ClientLockId]                    [char] (32) NULL,        -- client id who locks the camera
  [CCH_ChannelStatus]                   [int] NOT NULL,          -- indicates the current status of a channel
-- VCR operation
  [CCH_StartRecordExists]               [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCH_StartRecord]                     [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCH_StopRecordExists]                [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCH_StopRecord]                      [int] NOT NULL,
  [CCH_RewindExists]                    [int] NOT NULL,          -- 0 = No, 1 = yes, 2 = use default
  [CCH_Rewind]                          [int] NOT NULL,          -- 0 = no, 1 = yes, 2 = use default
  [CCH_PauseExists]                     [int] NOT NULL,          -- 0=no, 1=yes, 2=use default
  [CCH_Pause]                           [int] NOT NULL,          -- 0=no, 1=yes, 2=use default
  [CCH_FastForwardExists]               [int] NOT NULL,          -- 0=no, 1=yes, 2=use default
  [CCH_FastForward]                     [int] NOT NULL,          -- 0=no, 1=yes, 2=use default
  [CCH_StartPlaybackExists]             [int] NOT NULL,          -- 0=no, 1=yes, 2=use default
  [CCH_StartPlayback]                   [int] NOT NULL,          -- 0=no, 1=yes, 2=use default
  [CCH_StopPlaybackExists]              [int] NOT NULL,          -- 0=no, 1=yes, 2=use default
  [CCH_StopPlayback]                    [int] NOT NULL,          -- 0=no, 1=yes, 2=use default
-- Misc camera
  [CCH_ArchiveExportExists]             [int] not null,          -- 0=no, 1= yes, 2= use default
  [CCH_ArchiveExport]                   [int] not null,
  [CCH_SetPreAlarmExists]               [int] not null,
  [CCH_SetPreAlarm]                     [int] not null,
  [CCH_SetPostAlarmExists]              [int] not null,
  [CCH_SetPostAlarm]                    [int] not null,
  [CCH_SetVideoLossAlarmExists]         [int] not null,
  [CCH_SetVideoLossAlarm]               [int] not null,
  [CCH_SetMotionDetectionAlarmExists]   [int] not null,
  [CCH_SetMotionDetectionAlarm]         [int] not null,
  [CCH_SetFrameRateExists]              [int] not null,
  [CCH_SetFrameRate]                    [int] not null,
  [CCH_SetResolutionExists]             [int] not null,
  [CCH_SetResolution]                   [int] not null,
-- Camera positions
  [CCH_PresetExists]                    [int] not null,
```

```
    [CCH_PresetMax]                 [int] not null,                 -- number of presets for this cameara
    [CCH_PresetStopExists]          [int] not null,
    [CCH_PresetStop]                [int] not null,
    [CCH_PresetRecordExists]        [int] not null,
    [CCH_PresetRecord]              [int] not null,
    [CCH_PresetPlayExists]          [int] not null,
    [CCH_PresetPlay]                [int] not null,
    -- PT control
    [CCH_TiltExists]                [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_Tilt]                      [int] not null,
    [CCH_PanExists]                 [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_Pan]                       [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_StopAllPanTilt]            [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
-- Zoom and lense control
    [CCH_ZoomExists]                [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_Zoom]                      [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_FocusExists]               [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_Focus]                     [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_IrisExists]                [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_IrisAutomatic]             [int] NOT NULL,                 -- 0 = No, 1 = Yes
    [CCH_Iris]                      [int] NOT NULL,                 -- negative for close = No, positive for open
    [CCH_LensSpeed]                 [int] NOT NULL,                 -- Lens speed
    [CCH_LensSpeedMax]              [int] NOT NULL,                 -- Maximun lens speed, 1 = speed not variable
    [CCH_StopAllZoomFocusIris]      [int],                          -- 0 = No, 1 = Yes, 2 = Use default
-- other channel mainipulation command and status
    [CCH_Arm]                       [int] not null,                 -- 0=no, 1 = yes, 2 = use default
    [CCH_DisArm]                    [int] not null,
    [CCH_IsArmed]                   [int] not null,                 -- check arm status
    [CCH_WiperExists]               [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_Wiper]                     [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_WasherExists]              [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_Washer]                    [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_LightExists]               [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_Light]                     [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_StatusExists]              [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_AuxiliaryExists]           [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_AuxiliaryCount]            [int] NOT NULL,                 -- Number configured in database
    [CCH_AuxiliaryMax]              [int] NOT NULL,                 -- Number created in namespace, -1 = Use default
    [CCH_AuxiliaryPlayExists]       [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    [CCH_AuxiliaryStopExists]       [int] NOT NULL,                 -- 0 = No, 1 = Yes, 2 = Use default
    site                            CHAR(32) NOT NULL,              -- Site Name
    CCH_guid                        uniqueidentifier NOT NULL,      -- Global unique record identifier
    CCH_checksum                    INTEGER NOT NULL,               -- Record checksum
    upd_guid                        uniqueidentifier NOT NULL,      -- Guid for update operation
    upd_mode                        SMALLINT NOT NULL,              -- Update mode (0 - delete, 1 - edit, 2 - new)
    upd_timestamp                   DATETIME NOT NULL,              -- Date / Time of the update
    upd_checksum                    INTEGER NOT NULL,               -- Checksum
    upd_rowguid                     uniqueidentifier ROWGUIDCOL DEFAULT
                                        newid() UNIQUE NOT NULL,    -- Global unique record identifier
    [CCH_alarm_debounce_motion]     SMALLINT,                       -- alarm debounce time
    [CCH_alarm_debounce_behavior]   SMALLINT,                       -- alarm debounce time
    [CCH_ext_id_address]            [int] NULL,
    [CCH_ext_guid_address]          [uniqueidentifier] NULL,
    [CCH_device_ip_address]         [char](16) NULL,
    [CCH_ext_encoder]               [int] NULL,
    [CCH_xml_info]                  nvarchar(1024) NULL,
    CONSTRAINT [PK_AVChannel_save] PRIMARY KEY NONCLUSTERED
    (
                                    [CCH_ID], upd_mode desc, upd_guid
    )  ON [PRIMARY]
) ON [PRIMARY]
;
```

CREATE TABLE [AVDryContact]

```
(
  [ADC_ID]              [int] IDENTITY NOT FOR REPLICATION NOT NULL, -- Unique ID of DryContact record
  [ADC_ParentID]        [int] NOT NULL,                              -- ID of parent record
  [ADC_ServerID]        [int] NOT NULL,                              -- ID of server record
  [ADC_Description]     [char] (64) NOT NULL,                        -- Description
  [ADC_Partition]       [int] NOT NULL,                              -- Partition ID to which this DryContact belongs
  [ADC_Public]          [int] NOT NULL CONSTRAINT
                          [DF_AVDryContact_Public] DEFAULT (1),      -- 1 = DryContact is public, 0 = DryContact is private
  [ADC_Number]          SMALLINT,                                    -- dry contact number
  site                  CHAR(32) NOT NULL,                           -- Site Name
  ADC_guid              uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,                   -- Global unique record identifier
  ADC_checksum          INTEGER NOT NULL,                            -- Record checksum
  [ADC_xml_info]        nvarchar(1024) NULL,
  CONSTRAINT [PK_AVDryContact] PRIMARY KEY CLUSTERED
  (
                        [ADC_ID]
  )  ON [PRIMARY],
  CONSTRAINT [FK_AVDryContact_CCTVServer] FOREIGN KEY
  (
                        [ADC_ServerID]
  ) REFERENCES [CCTVServer] (
                        [CSV_ID]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_AVDryContact_partition] FOREIGN KEY
  (
                        [ADC_Partition]
  ) REFERENCES [partition] (
                        [part_number]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_DryContact_AVSite] FOREIGN KEY
  (
                        [ADC_ParentID]
  ) REFERENCES [AVSite] (
                        [CAV_ID]
  ) NOT FOR REPLICATION
) ON [PRIMARY]
;
```

CREATE TABLE [AVDryContact_save]

```
(
  [ADC_ID]              [int] NOT NULL,                              -- Unique ID of DryContact record
  [ADC_ParentID]        [int] NOT NULL,                              -- ID of parent record
  [ADC_ServerID]        [int] NOT NULL,                              -- ID of server record
  [ADC_Description]     [char] (64) NOT NULL,                        -- Description
  [ADC_Partition]       [int] NOT NULL,                              -- Partition ID to which this DryContact belongs
  [ADC_Public]          [int] NOT NULL,                              -- 1 = DryContact is public, 0 = DryContact is private
  [ADC_Number]          SMALLINT,                                    -- dry contact number
  site                  CHAR(32) NOT NULL,                           -- Site Name
  ADC_guid              uniqueidentifier NOT NULL,                   -- Global unique record identifier
  ADC_checksum          INTEGER NOT NULL,                            -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                   -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                           -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                           -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                            -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,                   -- Global unique record identifier
  [ADC_xml_info]        nvarchar(1024) NULL,
  CONSTRAINT [PK_AVDryContact_save] PRIMARY KEY NONCLUSTERED
  (
                        [ADC_ID], upd_mode desc, upd_guid
  )  ON [PRIMARY],
```

```
) ON [PRIMARY]
;
```

CREATE TABLE [AVMonitor]

```
(
  [AMO_ID]                          [int] IDENTITY NOT FOR REPLICATION
                                      NOT NULL,                                  -- Unique ID of monitor record
  [AMO_ParentID]                    [int] NOT NULL,                            -- ID of parent record
  [AMO_Name]                        [char] (64) NOT NULL,                      -- OPC name of monitor
  [AMO_ServerID]                    [int] NOT NULL,                            -- ID of server record
  [AMO_Exists]                      [int] NOT NULL CONSTRAINT
                                      [DF_AVMonitor_Exists] DEFAULT (1),       -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_Description]                 [char] (64) NOT NULL,                      -- Description
  [AMO_Partition]                   [int] NOT NULL,                            -- Partition ID to which this monitor belongs
  [AMO_Public]                      [int] NOT NULL CONSTRAINT
                                      [DF_AVMonitor_Public] DEFAULT (1),       -- 1 = monitor is public, 0 = monitor is private
  [AMO_GeneralString]               [char] (64) NULL,                          -- monitor annotation
  [AMO_SequenceExists]              [int] NOT NULL CONSTRAINT
                                      [DF_AVMonitor_SequenceExists] DEFAULT (2),       -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceCount]               [int] NOT NULL CONSTRAINT
                                      [DF_AVMonitor_SequenceCount] DEFAULT (0),        -- Number configured in database
  [AMO_SequenceMax]                 [int] NOT NULL CONSTRAINT
                                      [DF_AVMonitor_SequenceMax] DEFAULT ((-1)),       -- Number created in namespace,
                                                                                         -1 = Use default
  [AMO_SequencePlayExists]          [int] NOT NULL CONSTRAINT
                                      [DF_AVMonitor_SequencePlayExists] DEFAULT (2),   -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceRecordExists]        [int] NOT NULL CONSTRAINT
                                      [DF_AVMonitor_SequenceRecordExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceStopExists]          [int] NOT NULL CONSTRAINT
                                      [DF_AVMonitor_SequenceStopExists] DEFAULT (2),   -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequencePauseExists]         [int] NOT NULL CONSTRAINT
                                      [DF_AVMonitor_SequencePauseExists] DEFAULT (2),  -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceRestartExists]       [int] NOT NULL CONSTRAINT
                                      [DF_AVMonitor_SequenceRestartExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceForwardExists]       [int] NOT NULL CONSTRAINT
                                      [DF_AVMonitor_SequenceForwardExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceBackwardExists]      [int] NOT NULL CONSTRAINT
                                      [DF_AVMonitor_SequenceBackwardExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceStepForwardExists]   [int] NOT NULL CONSTRAINT
                                      [DF_AVMonitor_SequenceStepForwardExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceStepBackwardExists]  [int] NOT NULL CONSTRAINT
                                      [DF_AVMonitor_SequenceStepBackwardExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceCameraForwardExists] [int] NOT NULL CONSTRAINT
                                      [DF_AVMonitor_SequenceCameraForwardExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceCameraBackwardExists] [int] NOT NULL CONSTRAINT
                                      [DF_AVMonitor_SequenceCameraBackwardExists] DEFAULT (2),-- 0 = No, 1 = Yes, 2 = Use default
  site                              CHAR(32) NOT NULL,                         -- Site Name
  AMO_guid                          uniqueidentifier ROWGUIDCOL DEFAULT
                                      newid() UNIQUE NOT NULL,                 -- Global unique record identifier
  AMO_checksum                      INTEGER NOT NULL,                          -- Record checksum
  [AMO_ext_id_address]              [int] NOT NULL,
  [AMO_ext_guid_address]            [uniqueidentifier] NULL,
  [AMO_device_ip_address]           [char](16) NULL,
  [AMO_ext_decoder]                 [int] NOT NULL,
  [AMO_xml_info]                    nvarchar(1024) NULL,
  CONSTRAINT [PK_AVMonitor] PRIMARY KEY CLUSTERED
  (
                                    [AMO_ID]
  )  ON [PRIMARY],
  CONSTRAINT [FK_AVMonitor_CCTVServer] FOREIGN KEY
  (
                                    [AMO_ServerID]
  ) REFERENCES [CCTVServer] (
```

```
                                        [CSV_ID]
  )  NOT FOR REPLICATION,
  CONSTRAINT [FK_AVMonitor_partition] FOREIGN KEY
  (
                                        [AMO_Partition]
  ) REFERENCES [partition] (
                                        [part_number]
  )  NOT FOR REPLICATION,
  CONSTRAINT [FK_Monitor_AVSite] FOREIGN KEY
  (
                                        [AMO_ParentID]
  ) REFERENCES [AVSite] (
                                        [CAV_ID]
  ) NOT FOR REPLICATION
) ON [PRIMARY]
;
```

CREATE TABLE [AVMonitor_save]

```
(
  [AMO_ID]                          [int] NOT NULL,                          -- Unique ID of monitor record
  [AMO_ParentID]                    [int] NOT NULL,                          -- ID of parent record
  [AMO_Name]                        [char] (64) NOT NULL,                    -- OPC name of monitor
  [AMO_ServerID]                    [int] NOT NULL,                          -- ID of server record
  [AMO_Exists]                      [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_Description]                 [char] (64) NOT NULL,                    -- Description
  [AMO_Partition]                   [int] NOT NULL,                          -- Partition ID to which this monitor belongs
  [AMO_Public]                      [int] NOT NULL,                          -- 1 = monitor is public, 0 = monitor is private
  [AMO_GeneralString]               [char] (64) NULL,                        -- monitor annotation
  [AMO_SequenceExists]              [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceCount]               [int] NOT NULL,                          -- Number configured in database
  [AMO_SequenceMax]                 [int] NOT NULL,                          -- Number created in namespace, -1 = Use default
  [AMO_SequencePlayExists]          [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceRecordExists]        [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceStopExists]          [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequencePauseExists]         [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceRestartExists]       [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceForwardExists]       [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceBackwardExists]      [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceStepForwardExists]   [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceStepBackwardExists]  [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceCameraForwardExists] [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
  [AMO_SequenceCameraBackwardExists] [int] NOT NULL,                         -- 0 = No, 1 = Yes, 2 = Use default
  site                              CHAR(32) NOT NULL,                       -- Site Name
  AMO_guid                          uniqueidentifier NOT NULL,               -- Global unique record identifier
  AMO_checksum                      INTEGER NOT NULL,                        -- Record checksum
  upd_guid                          uniqueidentifier NOT NULL,               -- Guid for update operation
  upd_mode                          SMALLINT NOT NULL,                       -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                     DATETIME NOT NULL,                       -- Date / Time of the update
  upd_checksum                      INTEGER NOT NULL,                        -- Checksum
  upd_rowguid                       uniqueidentifier ROWGUIDCOL DEFAULT
                                      newid() UNIQUE NOT NULL,               -- Global unique record identifier
  [AMO_ext_id_address]              [int] NULL,
  [AMO_ext_guid_address]            [uniqueidentifier] NULL,
  [AMO_device_ip_address]           [char](16) NULL,
  [AMO_ext_decoder]                 [int] NULL,
  [AMO_xml_info]                    nvarchar(1024) NULL,
  CONSTRAINT [PK_AVMonitor_save] PRIMARY KEY NONCLUSTERED
  (
                                        [AMO_ID], upd_mode desc, upd_guid
  )  ON [PRIMARY],
) ON [PRIMARY]
;
```

CREATE TABLE [dbo].[AVPreset]

```
(
  [CAP_ID]              [int] IDENTITY NOT FOR REPLICATION NOT NULL,                -- Unique ID of preset record
  [CAP_ParentID]        [int] NOT NULL,                                             -- ID of parent record
  [CAP_Name]            [char] (64) NOT NULL,                                       -- OPC name of preset
  [CAP_ServerID]        [int] NOT NULL,                                             -- ID of server record
  [CAP_Exists]          [int] NOT NULL CONSTRAINT [DF_AVPreset_Exists] DEFAULT (1), -- 0 = No, 1 = Yes, 2 = Use default
  [CAP_Description]     [char] (64) NOT NULL,                                       -- Description
  [CAP_Partition]       [int] NOT NULL,                                             -- Partition ID to which this preset belongs
  [CAP_Public]          [int] NOT NULL,                                             -- 1 = preset is public, 0 = preset is private
  site                  CHAR(32) NOT NULL,                                          -- Site Name
  CAP_guid              uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,                                  -- Global unique record identifier
  CAP_checksum          INTEGER NOT NULL,                                           -- Record checksum
  [CAP_xml_info]        nvarchar(1024) NULL,
  CONSTRAINT [PK_AVPreset] PRIMARY KEY  CLUSTERED
  (
                        [CAP_ID]
  )  ON [PRIMARY],
  CONSTRAINT [FK_AVPreset_AVChannel] FOREIGN KEY
  (
                        [CAP_ParentID]
  ) REFERENCES [dbo].[AVChannel] (
                        [CCH_ID]
  )  NOT FOR REPLICATION,
  CONSTRAINT [FK_AVPreset_CCTVServer] FOREIGN KEY
  (
                        [CAP_ServerID]
  ) REFERENCES [dbo].[CCTVServer] (
                        [CSV_ID]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_AVPreset_partition] FOREIGN KEY
  (
                        [CAP_Partition]
  ) REFERENCES [dbo].[partition] (
                        [part_number]
  ) NOT FOR REPLICATION
) ON [PRIMARY]
;
```

CREATE TABLE [AVPreset_save]

```
(
  [CAP_ID]              [int]  NOT NULL,                                            -- Unique ID of preset record
  [CAP_ParentID]        [int] NOT NULL,                                             -- ID of parent record
  [CAP_Name]            [char] (64) NOT NULL,                                       -- OPC name of preset
  [CAP_ServerID]        [int] NOT NULL,                                             -- ID of server record
  [CAP_Exists]          [int] NOT NULL,                                             -- 0 = No, 1 = Yes, 2 = Use default
  [CAP_Description]     [char] (64) NOT NULL,                                       -- Description
  [CAP_Partition]       [int] NOT NULL,                                             -- Partition ID to which this preset belongs
  [CAP_Public]          [int] NOT NULL,                                             -- 1 = preset is public, 0 = preset is private
  site                  CHAR(32) NOT NULL,                                          -- Site Name
  CAP_guid              uniqueidentifier NOT NULL,                                  -- Global unique record identifier
  CAP_checksum          INTEGER NOT NULL,                                           -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                                  -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                          -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                           -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,                                  -- Global unique record identifier
  [CAP_xml_info]        nvarchar(1024) NULL,
  CONSTRAINT [PK_AVPreset_save] PRIMARY KEY NONCLUSTERED
  (
                        [CAP_ID], upd_mode desc, upd_guid
```

```
  ) ON [PRIMARY],
) ON [PRIMARY]
;
```

CREATE TABLE [AVSite]

```
(
  [CAV_ID]                    [int] IDENTITY NOT FOR REPLICATION NOT NULL,          -- Unique ID of switch record
  [CAV_ParentID]              [int] NOT NULL,                                       -- ID of parent record
  [CAV_Name]                  [char] (64) NOT NULL,                                 -- OPC name of switch
  [CAV_ServerID]              [int] NOT NULL,                                       -- ID of server record
  [CAV_Exists]                [int] NOT NULL CONSTRAINT
                                [DF_AVSite_Exists] DEFAULT (1),                     -- 0 = No, 1 = Yes, 2 = Use default
  [CAV_Description]           [char] (64) NOT NULL,                                 -- Description
  [CAV_Partition]             [int] NOT NULL CONSTRAINT
                                [DF_AVSite_Partition] DEFAULT (0),                  -- Partition ID to which this switch belongs
  [CAV_Public]                [int] NOT NULL CONSTRAINT
                                [DF_AVSite_Public] DEFAULT (1),                     -- 1 = switch is public, 0 = switch is private
  [CAV_Priority]              [int] NOT NULL CONSTRAINT
                                [DF_AVSite_Priority] DEFAULT (0),                   -- Not used
  [CAV_DVRType]               [char] (64) NOT NULL CONSTRAINT
                                [DF_AVSite_DVRType] DEFAULT (1),                    -- DVR manufacturer protocol type, 1 Loronix
  [CAV_Type]                  [int] NOT NULL CONSTRAINT
                                [DF_AVSite_Type] DEFAULT (0),                       -- 1 = Serial comms,
                                                                                      2 = TCP/IP, DVR communication type
  [CAV_DataServerIPAddress]   [char] (64) NOT NULL CONSTRAINT
                                [DF_AVSite_DataServerIPAddress] DEFAULT ('127.0.0.0'),          -- IP address of data server
  [CAV_MatrixServerIPAddress] [char] (64) NOT NULL CONSTRAINT
                                [DF_AVSite_MatrixServerIPAddress] DEFAULT ('127.0.0.0'),        -- IP address of data server
  [CAV_MatrixServerPort]      [char] (64) NOT NULL CONSTRAINT
                                [DF_AVSite_MatrixServerPort] DEFAULT ('COM1'),                  -- IP address of data server
  [CAV_MatrixServerName]      [char] (64) NOT NULL CONSTRAINT
                                [DF_AVSite_MatrixServerName] DEFAULT ('NONE'),                  -- IP address of data server
  [CAV_StreamingServerIPAddress] [char] (64) NOT NULL CONSTRAINT
                                [DF_AVSite_StreamingServerIPAddress] DEFAULT ('127.0.0.0'),     -- IP address of streaming server
  [CAV_StorageServerIPAddress] [char] (64) NOT NULL CONSTRAINT
                                [DF_AVSite_StorageServerIPAddress] DEFAULT ('127.0.0.0'),       -- IP address of storage server
  [CAV_LoginID]               [char] (64) NOT NULL CONSTRAINT
                                [DF_AVSite_LoginID] DEFAULT ('p2000'),              -- Login id on DVR
  [CAV_Password]              [char] (64) NOT NULL CONSTRAINT
                                [DF_AVSite_Password] DEFAULT ('letmein'),           -- login password on DVR, encrypted
  [CAV_NetBandwidth]          [int]  NOT NULL CONSTRAINT
                                [DF_AVSite_NetBandwidth] DEFAULT (2),               -- video transmission bandwidth
                                                                                      0:10MBPS, 1:100MBPS,
                                                                                      2:check at start up 3: 1GBPS,
  [CAV_ArchiveDeviceExists]   [int] NOT NULL CONSTRAINT
                                [DF_AVSite_ArchiveDeviceExists] DEFAULT (0),        -- no dat exists, 1:exists
  [CAV_ArchiveName]           [char] (64) NOT NULL CONSTRAINT
                                [DF_AVSite_ArchiveName] DEFAULT ('no_archive'),
  [CAV_ArchiveSize]           [int] NOT NULL CONSTRAINT
                                [DF_AVSite_ArchiveSize] DEFAULT (0),
  [CAV_NumArchiveDevices]     [int] NOT NULL CONSTRAINT
                                [DF_AVSite_NumArchiveDevices] default (0),          -- indication of a number of devs
  [CAV_ArchiveHasMedia]       [int] NOT NULL CONSTRAINT
                                [DF_AVSite_ArchiveHasMedia] default (0),            -- 1: has media, 0: no media in a deck
  [CAV_DateTime]              [int] NOT NULL CONSTRAINT
                                [DF_AVSite_DateTime] default (0),                   -- indidate that DVR's time has been
                                                                                      changed
  [CAV_AlarmClearAll]         [int] NOT null constraint
                                [DF_AVSite_AlarmClearAll] default (0),              -- alear all alarms
  [CAV_Login]                 [int] not null constraint[DF_AVSite_Login] default (0),  -- 1: logged on 0: not logged in
  [CAV_Logoff]                [int] not null constraint [DF_AVSite_Logoff] default (1), -- 1: not logged in, 0 : logged on
  [CAV_LoginState]            [int] not null constraint [DF_AVSite_LoginState] default (0), -- check do we have a pc
  [CAV_MimicDVR]              [int] not null constraint [DF_AVSite_MimicDVR] default (0),   -- mimic a DVR
```

```
[CAV_TestIP]                    [int] not null constraint [DF_AVSite_TestIP] default (0),                -- 1: request the validity of IP address of
                                                                                                            a dvr
[CAV_CheckPIN]                  [int] not null constraint [DF_AVSite_CheckPIN] default (0),              -- 1: check equipment
[CAV_ErrorSend]                 [int] not null constraint [DF_AVSite_ErrorSend] default (0),
[CAV_FatalErrorSend]            [int] not null constraint [DF_AVSite_FatalErrorSend] default (0),
[CAV_Special]                   [int] not null constraint [DF_AVSite_Special] default (0),
[CAV_GeneralString]             [char] (64) not null constraint [DF_AVSite_GeneralString] default ('dvr_comment'),
[CAV_PresetExists]              [int] not null constraint [DF_AVSite_PresetExists] default ((-1)),
[CAV_PresetStopExists]          [int] not null constraint [DF_AVSite_PresetStopExists] default ((-1)),
[CAV_PresetRecordExists]        [int] not null constraint [DF_AVSite_PresetRecordExists] default ((-1)),
[CAV_PresetPlayExists]          [int] not null constraint [DF_AVSite_PresetPlayExits] default ((-1)),
[CAV_CameraAuxiliaryExists]         [int] not null constraint [DF_AVSite_CameraAuxiliaryExists] default ((-1)),
[CAV_CameraAuxiliaryPlayExists]     [int] not null constraint [DF_AVSite_CameraAuxiliaryPlayExits] default ((-1)),
[CAV_CameraAuxiliaryStopExists]     [int] not null constraint [DF_AVSite_CameraAuxiliaryStopExists] default ((-1)),
[CAV_CameraAuxiliaryMax]            [int] not null constraint [DF_AVSite_CameraAuxilaryMax] default (0),
[CAV_PresetMax]                     [int] not null constraint [DF_AVSite_PresetMax]  default ((-1)),
[CAV_CameraGroup]                   [int] not null constraint [DF_AVSite_CameraGroup] default (0),
[CAV_CameraGroupMax]                [int] not null constraint [DF_AVSite_CameraGroupMax] default ((-1)),
[CAV_CameraCount]                   [int] not null constraint [DF_AVSite_CameraCount] default (0),
[CAV_CameraMax]                     [int] not null constraint [DF_AVSite_CameraMax] default ((-1)),
[CAV_MonitorCount]                  [int] NOT NULL CONSTRAINT [DF_AVSite_MonitorCount] DEFAULT (0),
[CAV_MonitorMax]                    [int] NOT NULL CONSTRAINT [DF_AVSite_MonitorMax] DEFAULT ((-1)),
[CAV_ActiveCameraExists]            [int] not null constraint [DF_AVSite_ActiveCameraExists] default (0),
[CAV_SetActiveCamera]               [int] not null constraint [DF_AVSite_SetActiveCamerea] default (0),
[CAV_ChannelGroup]              [int] not null constraint [DF_AVSite_ChannelGroup] default (0),          -- the number of camera group
[CAV_ChannelGroupMax]           [int] not null constraint [DF_AVSite_ChannelGroupMax] default (0),       -- the number of channel group max
[CAV_ChannelCount]              [int] not null constraint [DF_AVSite_ChannelCount] default (0),          -- number of channels configured
[CAV_ChannelMax]                [int] not null constraint [DF_AVSite_Channelmax] default ((-1)),         -- the number of channels configured in
                                                                                                            this DVR
[CAV_ActiveChannelExists]       [int] not null constraint [DF_AVSiteChannelExists] default (1),
[CAV_SetActiveChannel]          [int] not null constraint [DF_AVSite_SetActiveChannel] default (1),
[CAV_AlarmExists]                   [int] NOT NULL CONSTRAINT
                                        [DF_AVSite_AlarmExists] DEFAULT (2),                             -- 0 = No, 1 = Yes, 2 = Use default
[CAV_AlarmCount]                    [int] NOT NULL CONSTRAINT
                                        [DF_AVSite_AlarmCount] DEFAULT (0),                              -- Number configured in database
[CAV_AlarmMax]                      [int] NOT NULL CONSTRAINT
                                        [DF_AVSite_AlarmMax] DEFAULT ((-1)),                             -- Number created in namespace,
                                                                                                            -1 = Use default
[CAV_AlarmLossOfVideoSignalExists]  [int] NOT NULL CONSTRAINT
                                        [DF_AVSite_AlarmLossOfVideoSignalExists] DEFAULT (2),            -- 0 = No, 1 = Yes, 2 = Use default
[CAV_AlarmLossOfVideoSignal]        [int] NOT NULL CONSTRAINT
                                        [DF_AVSite_AlarmLossOfVideoSignal] DEFAULT (0),                  -- 0 = No, 1 = Yes, 2 = Use default
[CAV_AlarmStopLossOfVideoSignalExists] [int] NOT NULL CONSTRAINT
                                        [DF_AVSite_AlarmStopLossOfVideoSignalExists] DEFAULT (2),-- 0 = No, 1 = Yes, 2 = Use default
[CAV_AlarmMotionDetectionExists]    [int] NOT NULL CONSTRAINT
                                        [DF_AVSite_AlarmMotionDetectionExists] DEFAULT (2),              -- 0 = No, 1 = Yes, 2 = Use default
[CAV_AlarmMotionDetection]          [int] NOT NULL CONSTRAINT
                                        [DF_AVSite_AlarmMotionDetection] DEFAULT (0),                    -- 0 = No, 1 = Yes, 2 = Use default
[CAV_AlarmStopMotionDetection]      [int] NOT NULL CONSTRAINT
                                        [DF_AVSite_AlarmStopMotionDetection] DEFAULT (2),                -- 0 = No, 1 = Yes, 2 = Use default
[CAV_AuxiliaryExists]               [int] NOT NULL CONSTRAINT
                                        [DF_AVSite_AuxiliaryExists] DEFAULT (2),                         -- 0 = No, 1 = Yes, 2 = Use default
[CAV_AuxiliaryCount]                [int] NOT NULL CONSTRAINT
                                        [DF_AVSite_AuxiliaryCount] DEFAULT (0),                          -- Number configured in database
[CAV_AuxiliaryMax]                  [int] NOT NULL CONSTRAINT
                                        [DF_AVSite_AuxiliaryMax] DEFAULT ((-1)),                         -- Number created in namespace,
                                                                                                            -1 = Use default
[CAV_AuxiliaryPlayExists]           [int] NOT NULL CONSTRAINT
                                        [DF_AVSite_AuxiliaryPlayExists] DEFAULT (2),                     -- 0 = No, 1 = Yes, 2 = Use default
[CAV_AuxiliaryPlay]                 [int] NOT NULL CONSTRAINT
                                        [DF_AVSite_AuxiliaryPlay] DEFAULT (0),                           -- 0 = No, 1 = Yes, 2 = Use default
[CAV_AuxiliaryStopExists]           [int] NOT NULL CONSTRAINT
                                        [DF_AVSite_AuxiliaryStopExists] DEFAULT (2),                     -- 0 = No, 1 = Yes, 2 = Use default
[CAV_AuxiliaryStop]                 [int] NOT NULL CONSTRAINT
                                        [DF_AVSite_AuxiliaryStop] DEFAULT (0),                           -- 0 = No, 1 = Yes, 2 = Use default
```

```
   site                                CHAR(32) NOT NULL,                                  -- Site Name
   CAV_guid                            uniqueidentifier ROWGUIDCOL
                                          DEFAULT newid() UNIQUE  NOT NULL,                -- Global unique record identifier
   CAV_checksum                        INTEGER NOT NULL,                                   -- Record checksum
   [CAV_In_MotionTimezone]             INTEGER NULL REFERENCES timezone(tz_id)
                                          NOT FOR REPLICATION,                             -- time zone ID
   [CAV_In_BehaviorTimezone]           INTEGER NULL REFERENCES timezone(tz_id)
                                          NOT FOR REPLICATION,                             -- time zone ID
   [CAV_In_VideolossTimezone]          INTEGER NULL REFERENCES timezone(tz_id)
                                          NOT FOR REPLICATION,                             -- time zone ID
   [CAV_In_DrycontactTimezone]         INTEGER NULL REFERENCES timezone(tz_id)
                                          NOT FOR REPLICATION,                             -- time zone ID
   [CAV_Out_Mode]                      [int] NOT NULL,                                     -- 0 send no AV Alarms, 1 send all
                                                                                              AV Alarms, 2 send filtered
                                                                                              AV Alarms
   [CAV_Out_Timezone1]                 INTEGER NULL REFERENCES timezone(tz_id)
                                       NOT FOR REPLICATION,                                -- time zone ID
   [CAV_Out_Filtergroup1]                 INTEGER NULL REFERENCES msgfiltergroup(mfg_id) NOT FOR REPLICATION,--
message filter group ID
   [CAV_Motion_Debounce_Time]             [int] NOT NULL CONSTRAINT [DF_Motion_Debounce_Time] DEFAULT (1000),
   [CAV_Behavior_Debounce_Time]           [int] NOT NULL CONSTRAINT [DF_Behavior_Debounce_Time] DEFAULT (1000),
   [CAV_DryContactCount]                  [int] NOT NULL CONSTRAINT [DF_AVSite_DryContactCount] DEFAULT (0),
   [CAV_DryContactMax]                    [int] NOT NULL CONSTRAINT [DF_AVSite_DryContactMax] DEFAULT ((-1)),
   [CAV_primaryDirectoryServerIP_subnet1] [char] (64) NOT NULL,
   [CAV_secondaryDirectoryServerIP_subnet1] [char] (64) NOT NULL,
   [CAV_tertiaryDirectoryServerIP_subnet1] [char] (64) NOT NULL,
   [CAV_use_subnet2]                      SMALLINT NOT NULL,
   [CAV_primaryDirectoryServerIP_subnet2] [char] (64) NOT NULL,
   [CAV_secondaryDirectoryServerIP_subnet2] [char] (64) NOT NULL,
   [CAV_tertiaryDirectoryServerIP_subnet2] [char] (64) NOT NULL,
   [CAV_ext_id_address]                   [int] NOT NULL,
   [CAV_ext_guid_address]                 [uniqueidentifier] NULL,
   [CAV_xml_info]                         nvarchar(1024) NULL,
   CONSTRAINT [PK_AVSite] PRIMARY KEY CLUSTERED
   (
                              [CAV_ID]
   )  ON [PRIMARY],           --
   CONSTRAINT [FK_AVSite_CCTVServer] FOREIGN KEY
   (
                              [CAV_ServerID]
   ) REFERENCES [CCTVServer] (
                              [CSV_ID]
   ) NOT FOR REPLICATION,
   CONSTRAINT [FK_AVSite_partition] FOREIGN KEY
   (
                              [CAV_Partition]
   ) REFERENCES [partition] (
                              [part_number]
   ) NOT FOR REPLICATION,
) ON [PRIMARY]
;
```

CREATE TABLE [AVSite_save]

```
(
   [CAV_ID]                            [int] NOT NULL,                                     -- Unique ID of switch record
   [CAV_ParentID]                      [int] NOT NULL,                                     -- ID of parent record
   [CAV_Name]                          [char] (64) NOT NULL,                               -- OPC name of switch
   [CAV_ServerID]                      [int] NOT NULL,                                     -- ID of server record
   [CAV_Exists]                        [int] NOT NULL,                                     -- 0 = No, 1 = Yes, 2 = Use default
   [CAV_Description]                   [char] (64) NOT NULL,                               -- Description
   [CAV_Partition]                     [int] NOT NULL,                                     -- Partition ID to which this switch belongs
   [CAV_Public]                        [int] NOT NULL,                                     -- 1 = switch is public, 0 = switch is private
   [CAV_Priority]                      [int] NOT NULL,                                     -- Not used
```

[CAV_DVRType]	[char] (64) NOT NULL,	-- DVR manufacturer protocol type, 1 Loronix
[CAV_Type]	[int] NOT NULL,	-- 1 = Serial comms, 2 = TCP/IP, DVR communication type
[CAV_DataServerIPAddress]	[char] (64) NOT NULL,	-- IP address of data server
[CAV_MatrixServerIPAddress]	[char] (64) NOT NULL,	-- IP address of data server
[CAV_MatrixServerPort]	[char] (64) NOT NULL,	-- IP address of data server
[CAV_MatrixServerName]	[char] (64) NOT NULL,	-- IP address of data server
[CAV_StreamingServerIPAddress]	[char] (64) NOT NULL,	-- IP address of streaming server
[CAV_StorageServerIPAddress]	[char] (64) NOT NULL,	-- IP address of storage server
[CAV_LoginID]	[char] (64) NOT NULL,	-- Login id on DVR
[CAV_Password]	[char] (64) NOT NULL,	-- login password on DVR, encrypted
[CAV_NetBandwidth]	[int] NOT NULL,	-- video transmission bandwidth 0:10MBPS, 1:100MBPS, 2:check at start up 3: 1GBPS,
[CAV_ArchiveDeviceExists]	[int] NOT NULL,	-- no dat exists, 1:exists
[CAV_ArchiveName]	[char] (64) NOT NULL,	
[CAV_ArchiveSize]	[int] NOT NULL,	
[CAV_NumArchiveDevices]	[int] NOT NULL,	-- indication of a number of devs
[CAV_ArchiveHasMedia]	[int] NOT NULL,	-- 1: has media, 0: no media in a deck
[CAV_DateTime]	[int] NOT NULL,	-- indidate that DVR's time has been changed
[CAV_AlarmClearAll]	[int] NOT null,	-- alear all alarms
[CAV_Login]	[int] not null,	-- 1: logged on 0: not logged in
[CAV_Logoff]	[int] not null,	-- 1: not logged in, 0 : logged on
[CAV_LoginState]	[int] not null,	-- check do we have a pc
[CAV_MimicDVR]	[int] not null,	-- mimic a DVR
[CAV_TestIP]	[int] not null,	-- 1: request the validity of IP address of a dvr
[CAV_CheckPIN]	[int] not null,	-- 1: check equipment
[CAV_ErrorSend]	[int] not null,	
[CAV_FatalErrorSend]	[int] not null,	
[CAV_Special]	[int] not null,	
[CAV_GeneralString]	[char] (64) not null,	
[CAV_PresetExists]	[int] not null,	
[CAV_PresetStopExists]	[int] not null,	
[CAV_PresetRecordExists]	[int] not null,	
[CAV_PresetPlayExists]	[int] not null,	
[CAV_CameraAuxiliaryExists]	[int] not null,	
[CAV_CameraAuxiliaryPlayExists]	[int] not null,	
[CAV_CameraAuxiliaryStopExists]	[int] not null,	
[CAV_CameraAuxiliaryMax]	[int] not null,	
[CAV_PresetMax]	[int] not null,	
[CAV_CameraGroup]	[int] not null,	
[CAV_CameraGroupMax]	[int] not null,	
[CAV_CameraCount]	[int] not null,	
[CAV_CameraMax]	[int] not null,	
[CAV_MonitorCount]	[int] NOT NULL,	
[CAV_MonitorMax]	[int] NOT NULL,	
[CAV_ActiveCameraExists]	[int] not null,	
[CAV_SetActiveCamera]	[int] not null,	
[CAV_ChannelGroup]	[int] not null,	-- the number of camera group
[CAV_ChannelGroupMax]	[int] not null,	-- the number of channel group max
[CAV_ChannelCount]	[int] not null,	-- number of channels configured
[CAV_ChannelMax]	[int] not null,	-- the number of channels configured in this DVR
[CAV_ActiveChannelExists]	[int] not null,	
[CAV_SetActiveChannel]	[int] not null,	
[CAV_AlarmExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CAV_AlarmCount]	[int] NOT NULL,	-- Number configured in database
[CAV_AlarmMax]	[int] NOT NULL,	-- Number created in namespace, -1 = Use default
[CAV_AlarmLossOfVideoSignalExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CAV_AlarmLossOfVideoSignal]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CAV_AlarmStopLossOfVideoSignalExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CAV_AlarmMotionDetectionExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CAV_AlarmMotionDetection]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CAV_AlarmStopMotionDetection]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CAV_AuxiliaryExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CAV_AuxiliaryCount]	[int] NOT NULL,	-- Number configured in database
[CAV_AuxiliaryMax]	[int] NOT NULL,	-- Number created in namespace, -1 = Use default
[CAV_AuxiliaryPlayExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default

```
  [CAV_AuxiliaryPlay]                [int] NOT NULL,                -- 0 = No, 1 = Yes, 2 = Use default
  [CAV_AuxiliaryStopExists]          [int] NOT NULL,                -- 0 = No, 1 = Yes, 2 = Use default
  [CAV_AuxiliaryStop]                [int] NOT NULL,                -- 0 = No, 1 = Yes, 2 = Use default
  site                               CHAR(32) NOT NULL,             -- Site Name
  CAV_guid                           uniqueidentifier NOT NULL,     -- Global unique record identifier
  CAV_checksum                       INTEGER NOT NULL,              -- Record checksum
  upd_guid                           uniqueidentifier NOT NULL,     -- Guid for update operation
  upd_mode                           SMALLINT NOT NULL,             -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                      DATETIME NOT NULL,             -- Date / Time of the update
  upd_checksum                       INTEGER NOT NULL,              -- Checksum
  upd_rowguid                        uniqueidentifier ROWGUIDCOL DEFAULT
                                       newid() UNIQUE NOT NULL,     -- Global unique record identifier
  -- dvr alarm and message fordwarding (p2000 3.1)
  [CAV_In_MotionTimezone]            INTEGER NULL,                  -- time zone ID
  [CAV_In_BehaviorTimezone]          INTEGER NULL,                  -- time zone ID
  [CAV_In_VideolossTimezone]         INTEGER NULL,                  -- time zone ID
  [CAV_In_DrycontactTimezone]        INTEGER NULL,                  -- time zone ID
  [CAV_Out_Mode]                     [int] NOT NULL,                -- 0 send no AV Alarms, 1 send all AV Alarms,
                                                                       2 send filtered AV Alarms
  [CAV_Out_Timezone1]                INTEGER NULL,                  -- time zone ID
  [CAV_Out_Filtergroup1]             INTEGER NULL,                  -- message filter group ID
  [CAV_Motion_Debounce_Time]         [int] NOT NULL,
  [CAV_Behavior_Debounce_Time]       [int] NOT NULL,
  [CAV_DryContactCount]              [int] NOT NULL,
  [CAV_DryContactMax]                [int] NOT NULL,
  [CAV_primaryDirectoryServerIP_subnet1]   [char] (64) NOT NULL,
  [CAV_secondaryDirectoryServerIP_subnet1] [char] (64) NOT NULL,
  [CAV_tertiaryDirectoryServerIP_subnet1]  [char] (64) NOT NULL,
  [CAV_use_subnet2]                  SMALLINT,
  [CAV_primaryDirectoryServerIP_subnet2]   [char] (64) NOT NULL,
  [CAV_secondaryDirectoryServerIP_subnet2] [char] (64) NOT NULL,
  [CAV_tertiaryDirectoryServerIP_subnet2]  [char] (64) NOT NULL,
  [CAV_ext_id_address]               [int] NULL,
  [CAV_ext_guid_address]             [uniqueidentifier] NULL,
  [CAV_xml_info]                     nvarchar(1024) NULL,
  PRIMARY KEY                        ( CAV_Partition, CAV_ID, upd_mode desc, upd_guid )
);
```

CREATE TABLE bacnetoid

```
(
  bno_oid              INTEGER PRIMARY KEY,          -- unique BACnet object ID (also used as database ID)
  bno_type             SMALLINT,                     -- 0 = Counter, 1 = Reader Terminal, 2 = Input Terminal,
                                                        3 = Output Terminal, 4 = Reader, 5 = Input,  6 = Output,
                                                        7 = Host Event Notify, 8 = Host Log Notify,
                                                        9 = Host Logic Notify, 10 = Audit Log Notify,
                                                        11 = Panel Notify, 12 = Panel Event Notify,
                                                        13 = Hardware Notify, 14 = Terminal Notify,
                                                        15 = Reader Notify, 16 = Input Notify, 17 = Output Notify,
                                                        18 = Grant Notify, 19 = Deny Notify, 20 = Trace Notify,
                                                        21 = Time and Attendance Notify,
  bno_panel_id         INTEGER REFERENCES panel(p_panel_id)
                         NOT FOR REPLICATION,        -- unique ID of panel this Object ID references (only valid when
                                                        bno_type is 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 or 21)
  bno_counter_id       INTEGER REFERENCES counter(ctr_id)
                         NOT FOR REPLICATION,        -- unique ID of counter this Object ID references (only valid
                                                        when bno_type is 0)
  bno_terminal_id      INTEGER REFERENCES terminal(tp_term_id)
                         NOT FOR REPLICATION,        -- unique ID of terminal this Object ID references (only valid
                                                        when bno_type is 1, 2, 3 or 4)
  bno_input_id         INTEGER REFERENCES input(ip_point_id)
                         NOT FOR REPLICATION,        -- unique ID of input point this Object ID references (only valid
                                                        when bno_type is 5)
  bno_output_id        INTEGER REFERENCES output(op_output_id)
```

```
                    NOT FOR REPLICATION,                    -- unique ID of input point this Object ID references (only valid
                                                               when bno_type is 6)
  site              CHAR(32) NOT NULL,                      -- Site Name
  bno_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  bno_checksum      INTEGER NOT NULL,                       -- Record checksum
);
  CREATE INDEX      bno_type ON bacnetoid ( bno_type );
```

CREATE TABLE bacnetsource

```
(
  bns_id            INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,  -- unique database ID
  bns_name          CHAR(32) NOT NULL,                      -- unique Pegasys name of the source
  bns_ip_address    CHAR(16),                               -- IP address of the source (valid only if
                                                               bns_computer_name is empty)
  bns_computer_name CHAR(32),                               -- DNS name of the source (valid only if bns_ip_address
                                                               is empty)
  bns_allow         SMALLINT NOT NULL,                      -- 0 = Deny access, 1 = Allow access
  bns_routed        SMALLINT NOT NULL,                      -- 0 = not routed, 1 = Routed
  site              CHAR(32) NOT NULL,                      -- Site Name
  bns_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  bns_checksum      INTEGER NOT NULL,                       -- Record checksum
  PRIMARY KEY       ( bns_id )
);
  CREATE UNIQUE INDEX   bns_site_name ON bacnetsource (site, bns_name);
```

CREATE TABLE bacnetsource_save

```
(
  bns_id            INTEGER NOT NULL,                       -- unique database ID
  bns_name          CHAR(32) NOT NULL,                      -- unique Pegasys name of the source
  bns_ip_address    CHAR(16),                               -- IP address of the source (valid only if bns_computer_name is empty)
  bns_computer_name CHAR(32),                               -- DNS name of the source (valid only if bns_ip_address is empty)
  bns_allow         SMALLINT NOT NULL,                      -- 0 = Deny access, 1 = Allow access
  bns_routed        SMALLINT NOT NULL,                      -- 0 = not routed, 1 = Routed
  site              CHAR(32) NOT NULL,                      -- Site Name
  bns_guid          uniqueidentifier NOT NULL,              -- Global unique record identifier
  bns_checksum      INTEGER NOT NULL,                       -- Record checksum
  upd_guid          uniqueidentifier NOT NULL,              -- Guid for update operation
  upd_mode          SMALLINT NOT NULL,                      -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp     DATETIME NOT NULL,                      -- Date / Time of the update
  upd_checksum      INTEGER NOT NULL,                       -- Checksum
  upd_rowguid       uniqueidentifier ROWGUIDCOL DEFAULT
                      newid() UNIQUE NOT NULL,              -- Global unique record identifier
  PRIMARY KEY       ( bns_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE badge

```
(
  b_partition       INTEGER REFERENCES partition(part_number)
                      NOT FOR REPLICATION,                  -- partition ID to which this badge belongs
  b_public          SMALLINT,                               -- 1 = badge is public, 0 = badge is private
  b_number_str      VARCHAR(100) PRIMARY KEY NOT NULL,      -- badge number
  b_cardholder_id   INTEGER REFERENCES cardholder(c_id)
                      NOT FOR REPLICATION,                  -- cardholder ID of cardholder who has this badge
  b_alpha           CHAR(4) NULL,                           -- alpha field of badge
  b_issue           SMALLINT,                               -- issue level
  b_description     CHAR(64) NULL,                          -- user entered description
  b_disabled        SMALLINT NOT NULL,                      -- 1 = disabled, 0 = active
  b_exp_timestamp   DATETIME,                               -- expiration timestamp
  b_start_timestamp DATETIME,                               -- start timestamp
  b_pin             INTEGER,                                -- pin code
  b_event_priv      SMALLINT,                               -- event privilege
```

```
b_exec_priv         SMALLINT,                                                       -- 1 = executive privilege
b_trace             SMALLINT,                                                       -- 1 = card trace on
b_override          SMALLINT,                                                       -- 1 = override on
b_access_grp_0      INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 1
b_access_grp_1      INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 2
b_access_grp_2      INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 3
b_access_grp_3      INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 4
b_access_grp_4      INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 5
b_access_grp_5      INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 6
b_access_grp_6      INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 7
b_access_grp_7      INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 8
b_access_tz_0       INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,     -- access timezone # 1
b_access_tz_1       INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,     -- access timezone # 2
b_access_tz_2       INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,     -- access timezone # 3
b_access_tz_3       INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,     -- access timezone # 4
b_access_tz_4       INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,     -- access timezone # 5
b_access_tz_5       INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,     -- access timezone # 6
b_access_tz_6       INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,     -- access timezone # 7
b_access_tz_7       INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,     -- access timezone # 8
b_reason            CHAR(32) NULL,                                                  -- reason for badge issue
b_design            INTEGER NULL REFERENCES badgelayout(bl_id) NOT FOR REPLICATION,  -- badge design ID
b_interlock1        INTEGER REFERENCES actioninterlock(ai_id) NOT FOR REPLICATION,   -- id of action interlock 1 to activate
b_interlock1_value  FLOAT,                                                          -- value for action interlock 1
b_interlock2        INTEGER REFERENCES actioninterlock(ai_id) NOT FOR REPLICATION,   -- id of action interlock 2 to activate
b_interlock2_value  FLOAT,                                                          -- value for action interlock 2
b_priority          SMALLINT,                                                       -- Badge Priority
b_dnld_sti_e        SMALLINT,                                                       -- Download badge to STI-E if 1,
                                                                                       otherwise 0
b_ag_start0         DATETIME,                                                       -- start timestamp of access group 1
b_ag_start1         DATETIME,                                                       -- start timestamp of access group 2
b_ag_start2         DATETIME,                                                       -- start timestamp of access group 3
b_ag_start3         DATETIME,                                                       -- start timestamp of access group 4
b_ag_start4         DATETIME,                                                       -- start timestamp of access group 5
b_ag_start5         DATETIME,                                                       -- start timestamp of access group 6
b_ag_start6         DATETIME,                                                       -- start timestamp of access group 7
b_ag_start7         DATETIME,                                                       -- start timestamp of access group 8
b_ag_end0           DATETIME,                                                       -- end timestamp of access group 1
b_ag_end1           DATETIME,                                                       -- end timestamp of access group 2
b_ag_end2           DATETIME,                                                       -- end timestamp of access group 3
b_ag_end3           DATETIME,                                                       -- end timestamp of access group 4
b_ag_end4           DATETIME,                                                       -- end timestamp of access group 5
b_ag_end5           DATETIME,                                                       -- end timestamp of access group 6
b_ag_end6           DATETIME,                                                       -- end timestamp of access group 7
b_ag_end7           DATETIME,                                                       -- end timestamp of access group 8
b_facility_code     INTEGER,                                                        -- facility code 0 to 65535
b_security_level    SMALLINT,                                                       -- security level 0-99
b_flag_a            SMALLINT,                                                       -- special flag A
b_flag_b            SMALLINT,                                                       -- special flag B
b_flag_c            SMALLINT,                                                       -- special flag C
b_block_xaction     SMALLINT,                                                       -- Block Bages: 1= block transaction,
                                                                                       0 = allow transaction
site                CHAR(32) NOT NULL,                                              -- Site Name
site                CHAR(32) NOT NULL,                                              -- Site Name
b_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                    -- Global unique record identifier
b_checksum          INTEGER NOT NULL,                                               -- Record checksum
b_access_grp_8      INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 9
b_access_grp_9      INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 10
b_access_grp_10     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 11
b_access_grp_11     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 12
b_access_grp_12     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 13
b_access_grp_13     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 14
b_access_grp_14     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 15
b_access_grp_15     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 16
b_access_grp_16     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 17
b_access_grp_17     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,     -- access group # 18
```

```
b_access_grp_18     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 19
b_access_grp_19     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 20
b_access_grp_20     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 21
b_access_grp_21     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 22
b_access_grp_22     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 23
b_access_grp_23     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 24
b_access_grp_24     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 25
b_access_grp_25     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 26
b_access_grp_26     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 27
b_access_grp_27     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 28
b_access_grp_28     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 29
b_access_grp_29     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 30
b_access_grp_30     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 31
b_access_grp_31     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 32
b_access_tz_8       INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 9
b_access_tz_9       INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 10
b_access_tz_10      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 11
b_access_tz_11      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 12
b_access_tz_12      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 13
b_access_tz_13      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 14
b_access_tz_14      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 15
b_access_tz_15      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 16
b_access_tz_16      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 17
b_access_tz_17      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 18
b_access_tz_18      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 19
b_access_tz_19      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 20
b_access_tz_20      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 21
b_access_tz_21      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 22
b_access_tz_22      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 23
b_access_tz_23      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 24
b_access_tz_24      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 25
b_access_tz_25      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 26
b_access_tz_26      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 27
b_access_tz_27      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 28
b_access_tz_28      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 29
b_access_tz_29      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 30
b_access_tz_30      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 31
b_access_tz_31      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 32
b_ag_start8         DATETIME,                                                       -- start timestamp of access group 9
b_ag_start9         DATETIME,                                                       -- start timestamp of access group 10
b_ag_start10        DATETIME,                                                       -- start timestamp of access group 11
b_ag_start11        DATETIME,                                                       -- start timestamp of access group 12
b_ag_start12        DATETIME,                                                       -- start timestamp of access group 13
b_ag_start13        DATETIME,                                                       -- start timestamp of access group 14
b_ag_start14        DATETIME,                                                       -- start timestamp of access group 15
b_ag_start15        DATETIME,                                                       -- start timestamp of access group 16
b_ag_start16        DATETIME,                                                       -- start timestamp of access group 17
b_ag_start17        DATETIME,                                                       -- start timestamp of access group 18
b_ag_start18        DATETIME,                                                       -- start timestamp of access group 19
b_ag_start19        DATETIME,                                                       -- start timestamp of access group 20
b_ag_start20        DATETIME,                                                       -- start timestamp of access group 21
b_ag_start21        DATETIME,                                                       -- start timestamp of access group 22
b_ag_start22        DATETIME,                                                       -- start timestamp of access group 23
b_ag_start23        DATETIME,                                                       -- start timestamp of access group 24
b_ag_start24        DATETIME,                                                       -- start timestamp of access group 25
b_ag_start25        DATETIME,                                                       -- start timestamp of access group 26
b_ag_start26        DATETIME,                                                       -- start timestamp of access group 27
b_ag_start27        DATETIME,                                                       -- start timestamp of access group 28
b_ag_start28        DATETIME,                                                       -- start timestamp of access group 29
b_ag_start29        DATETIME,                                                       -- start timestamp of access group 30
b_ag_start30        DATETIME,                                                       -- start timestamp of access group 31
b_ag_start31        DATETIME,                                                       -- start timestamp of access group 32
b_ag_end8           DATETIME,                                                       -- end timestamp of access group 9
b_ag_end9           DATETIME,                                                       -- end timestamp of access group 10
b_ag_end10          DATETIME,                                                       -- end timestamp of access group 11
```

```
    b_ag_end11              DATETIME,                                                   -- end timestamp of access group 12
    b_ag_end12              DATETIME,                                                   -- end timestamp of access group 13
    b_ag_end13              DATETIME,                                                   -- end timestamp of access group 14
    b_ag_end14              DATETIME,                                                   -- end timestamp of access group 15
    b_ag_end15              DATETIME,                                                   -- end timestamp of access group 16
    b_ag_end16              DATETIME,                                                   -- end timestamp of access group 17
    b_ag_end17              DATETIME,                                                   -- end timestamp of access group 18
    b_ag_end18              DATETIME,                                                   -- end timestamp of access group 19
    b_ag_end19              DATETIME,                                                   -- end timestamp of access group 20
    b_ag_end20              DATETIME,                                                   -- end timestamp of access group 21
    b_ag_end21              DATETIME,                                                   -- end timestamp of access group 22
    b_ag_end22              DATETIME,                                                   -- end timestamp of access group 23
    b_ag_end23              DATETIME,                                                   -- end timestamp of access group 24
    b_ag_end24              DATETIME,                                                   -- end timestamp of access group 25
    b_ag_end25              DATETIME,                                                   -- end timestamp of access group 26
    b_ag_end26              DATETIME,                                                   -- end timestamp of access group 27
    b_ag_end27              DATETIME,                                                   -- end timestamp of access group 28
    b_ag_end28              DATETIME,                                                   -- end timestamp of access group 29
    b_ag_end29              DATETIME,                                                   -- end timestamp of access group 30
    b_ag_end30              DATETIME,                                                   -- end timestamp of access group 31
    b_ag_end31              DATETIME,                                                   -- end timestamp of access group 32
    b_purpose               uniqueidentifier REFERENCES badgepurpose(bp_guid) NOT FOR REPLICATION,
    b_data_style            SMALLINT NOT NULL,                                          -- 0 = normal, 1 = FASCN
    b_default_floor         SMALLINT,                                                   -- Otis Compass default floor for this
                                                                                           badge
    b_otis_pin_floor_mask   INTEGER REFERENCES floormask(fm_id)  NOT FOR REPLICATION,   -- id of floor mask used with Otis
                                                                                           Compass pin code is received
    b_operator              CHAR(32),                                                -- Operator that last modified record
    b_edit_time             DATETIME,                                                -- Time this record was last edited
    b_technology            INTEGER NOT NULL,                                        -- technology of badge format
    b_bits                  INTEGER NOT NULL,                                        -- the number of bits in the badge format
    b_qualifier             INTEGER NOT NULL,                                        -- format qualifier for the badge format
);
    CREATE INDEX            b_partition ON badge ( b_partition );
    CREATE INDEX            b_public ON badge ( b_public );
    CREATE INDEX            b_cardholder_id ON badge ( b_cardholder_id );
    CREATE INDEX            b_disabled ON badge ( b_disabled );
    CREATE INDEX            b_expire ON badge ( b_exp_timestamp );
    CREATE INDEX            b_valid ON badge ( b_start_timestamp );
    CREATE INDEX            b_ag_start0 ON badge ( b_ag_start0 );
    CREATE INDEX            b_ag_start1 ON badge ( b_ag_start1 );
    CREATE INDEX            b_ag_start2 ON badge ( b_ag_start2 );
    CREATE INDEX            b_ag_start3 ON badge ( b_ag_start3 );
    CREATE INDEX            b_ag_start4 ON badge ( b_ag_start4 );
    CREATE INDEX            b_ag_start5 ON badge ( b_ag_start5 );
    CREATE INDEX            b_ag_start6 ON badge ( b_ag_start6 );
    CREATE INDEX            b_ag_start7 ON badge ( b_ag_start7 );
    CREATE INDEX            b_ag_end0 ON badge ( b_ag_end0 );
    CREATE INDEX            b_ag_end1 ON badge ( b_ag_end1 );
    CREATE INDEX            b_ag_end2 ON badge ( b_ag_end2 );
    CREATE INDEX            b_ag_end3 ON badge ( b_ag_end3 );
    CREATE INDEX            b_ag_end4 ON badge ( b_ag_end4 );
    CREATE INDEX            b_ag_end5 ON badge ( b_ag_end5 );
    CREATE INDEX            b_ag_end6 ON badge ( b_ag_end6 );
    CREATE INDEX            b_ag_end7 ON badge ( b_ag_end7 );
```

CREATE TABLE badgefields

```
(
    bf_id                   INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,    -- unique database ID
    bf_name                 CHAR(60) NOT NULL,                              -- badge field's name
    bf_db_field_name        CHAR(30) NULL,                                  -- database field where this data comes from
    bf_db_table_name        CHAR(30) NOT NULL,                              -- database table where this data comes from
    bf_db_column_width      INTEGER NOT NULL,                               -- database field column width where this data comes
```

```
                                                                    from (ie maximum length of the data)
  bf_default_pad_char     CHAR(1) NOT NULL,                         -- For mag stripes, the pad character (ie <space> for text
                                                                       and zero for number)
  site                    CHAR(32) NOT NULL,                        -- Site Name
  bf_guid                 uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  bf_checksum             INTEGER NOT NULL,                         -- Record checksum
  PRIMARY KEY             ( bf_id, bf_name)
);
  CREATE UNIQUE INDEX     bf_site_name ON badgefields (site, bf_name);
```

CREATE TABLE badgefields_save

```
(
  bf_id                   INTEGER NOT NULL,                         -- unique database ID
  bf_name                 CHAR(60) NOT NULL,                        -- badge field's name
  bf_db_field_name        CHAR(30) NULL,                            -- database field where this data comes from
  bf_db_table_name        CHAR(30) NOT NULL,                        -- database table where this data comes from
  bf_db_column_width      INTEGER NOT NULL,                         -- database field column width where this data comes from (ie
                                                                       maximum length of the data)
  bf_default_pad_char     CHAR(1) NOT NULL,                         -- For mag stripes, the pad character (ie <space> for text and
                                                                       zero for number)
  site                    CHAR(32) NOT NULL,                        -- Site Name
  bf_guid                 uniqueidentifier NOT NULL,                -- Global unique record identifier
  bf_checksum             INTEGER NOT NULL,                         -- Record checksum
  upd_guid                uniqueidentifier NOT NULL,                -- Guid for update operation
  upd_mode                SMALLINT NOT NULL,                        -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp           DATETIME NOT NULL,                        -- Date / Time of the update
  upd_checksum            INTEGER NOT NULL,                         -- Checksum
  upd_rowguid             uniqueidentifier ROWGUIDCOL DEFAULT
                            newid() UNIQUE NOT NULL,                -- Global unique record identifier
  PRIMARY KEY             ( bf_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE badgeformats

```
(
  bf_name                 NVARCHAR(32) NOT NULL,                    -- user assigned name for badge format
  bf_technology           INTEGER NOT NULL,                         -- badge technology (0 = not specified, 1 = MagStripe,
                                                                       2 = Weigand, 3 = Prox, 4 = Corp1000, 5 = SmartCard,
                                                                       6 = IClass, 7 = Mifare)
  bf_bits                 INTEGER NOT NULL,                         -- number of bits in badge format
  bf_qualifier            INTEGER NOT NULL,                         -- qualifier of badge format
  bf_partition            INTEGER REFERENCES partition(part_number)
                            NOT FOR REPLICATION,                    -- partition ID to which this format belongs
  bf_public               SMALLINT,                                 -- 1 = format is public, 0 = format is private
  site                    NVARCHAR(32),                             -- Site Name
  bf_guid                 uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  bf_checksum             INTEGER NOT NULL,                         -- Record checksum
  PRIMARY KEY             (bf_guid)
);
```

CREATE TABLE badgeformats_save

```
(
  bf_name                 NVARCHAR(32) NOT NULL,                    -- user assigned name for badge format
  bf_technology           INTEGER NOT NULL,                         -- badge technology (0 = not specified, 1 = MagStripe,
                                                                       2 = Weigand, 3 = Prox, 4 = Corp1000, 5 = SmartCard,
                                                                       6 = IClass, 7 = Mifare)
  bf_bits                 INTEGER NOT NULL,                         -- number of bits in badge format
  bf_qualifier            INTEGER NOT NULL,                         -- qualifier of badge format
  bf_partition            INTEGER,                                  -- partition ID to which this format belongs
  bf_public               SMALLINT,                                 -- 1 = format is public, 0 = format is private
  site                    NVARCHAR(32) NOT NULL,                    -- Site Name
  bf_guid                 uniqueidentifier NOT NULL,                -- Global unique record identifier
```

```
  bf_checksum          INTEGER NOT NULL,                    -- Record checksum
  upd_guid             uniqueidentifier NOT NULL,           -- Guid for update operation
  upd_mode             SMALLINT NOT NULL,                   -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp        DATETIME NOT NULL,                   -- Date / Time of the update
  upd_checksum         INTEGER NOT NULL,                    -- Checksum
  PRIMARY KEY          (bf_guid, upd_mode desc, upd_guid)
);
```

CREATE TABLE badgelayout

```
(
  bl_id                INTEGER IDENTITY NOT FOR REPLICATION PRIMARY KEY,   -- unique database ID
  bl_name              CHAR(25) NOT NULL,                   -- badge layout name
  site                 CHAR(32) NOT NULL,                   -- Site Name
  bl_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  bl_checksum          INTEGER NOT NULL,                    -- Record checksum
  bl_image             IMAGE,                               -- layout image
);
  CREATE UNIQUE INDEX      bl_site_name ON badgelayout (site, bl_name);
```

CREATE TABLE badgelayout_save

```
(
  bl_id                INTEGER NOT NULL,                    -- unique database ID
  bl_name              CHAR(25) NOT NULL,                   -- badge layout name
  site                 CHAR(32) NOT NULL,                   -- Site Name
  bl_guid              uniqueidentifier NOT NULL,           -- Global unique record identifier
  bl_checksum          INTEGER NOT NULL,                    -- Record checksum
  upd_guid             uniqueidentifier NOT NULL,           -- Guid for update operation
  upd_mode             SMALLINT NOT NULL,                   -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp        DATETIME NOT NULL,                   -- Date / Time of the update
  upd_checksum         INTEGER NOT NULL,                    -- Checksum
  upd_rowguid          uniqueidentifier ROWGUIDCOL DEFAULT
                         newid() UNIQUE NOT NULL,           -- Global unique record identifier
  bl_image             IMAGE,                               -- layout image
  PRIMARY KEY          (bl_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE badgemagformula

```
(
  bmf_id               INTEGER IDENTITY NOT FOR
                         REPLICATION UNIQUE,                -- unique database ID
  bmf_formula_no       SMALLINT,                            -- mag stripe number, 1 = stripe 1, 2 = stripe 2, etc.
  bmf_field_type       SMALLINT,                            -- 0 = database field (includes UDF), 1 = user specified
  -- bmf_field_id      INTEGER NULL REFERENCES badgefields(bf_id),   -- Causes GPF in SQL Server when inserting a NULL value
                                                                      in this field (SQL Server 7 SP1)
  bmf_field_id         INTEGER NULL,                        -- field id from badge fields, NULL if field is user specified
  bmf_min_field_length SMALLINT,                            -- minimum length of data allowed in the field
  bmf_max_field_length SMALLINT,                            -- maximum length of data allowed in the field
  bmf_field_order      SMALLINT,                            -- order the fields appear on the stripe
  bmf_field_pad        CHAR(1) NOT NULL,                    -- character used between fields for padding
  bmf_field_content    CHAR(255) NULL,                      -- field name if this is a database field, actual data if this is a
                                                                user specified field
  bmf_right_pad        SMALLINT,                            -- 0 = Left pad string, 1 = Right pad string
  bmf_field_delimiter  CHAR(1) NULL,                        -- character used between fields for padding
  site                 CHAR(32) NOT NULL,                   -- Site Name
  bmf_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  bmf_checksum         INTEGER NOT NULL,                    -- Record checksum
  PRIMARY KEY          (bmf_id, bmf_formula_no)
);
```

CREATE TABLE badgemagformula_save

```
(
  bmf_id                INTEGER NOT NULL,                          -- unique database ID
  bmf_formula_no        SMALLINT,                                  -- mag stripe number, 1 = stripe 1, 2 = stripe 2, etc.
  bmf_field_type        SMALLINT,                                  -- 0 = database field (includes UDF), 1 = user specified
  bmf_field_id          INTEGER NULL,                              -- field id from badge fields, NULL if field is user specified
  bmf_min_field_length  SMALLINT,                                  -- minimum length of data allowed in the field
  bmf_max_field_length  SMALLINT,                                  -- maximum length of data allowed in the field
  bmf_field_order       SMALLINT,                                  -- order the fields appear on the stripe
  bmf_field_pad         CHAR(1) NOT NULL,                          -- character used between fields for padding
  bmf_field_content     CHAR(255) NULL,                            -- field name if this is a database field, actual data if this is a user
                                                                      specified field
  bmf_right_pad         SMALLINT,                                  -- 0 = Left pad string, 1 = Right pad string
  bmf_field_delimiter   CHAR(1) NULL,                              -- character used between fields for padding
  site                  CHAR(32) NOT NULL,                         -- Site Name
  bmf_guid              uniqueidentifier NOT NULL,                 -- Global unique record identifier
  bmf_checksum          INTEGER NOT NULL,                          -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                 -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                         -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                          -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,                 -- Global unique record identifier
  PRIMARY KEY           (bmf_id, bmf_formula_no, upd_mode desc, upd_guid)
);
```

CREATE TABLE badgepurpose

```
(
  bp_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  bp_partition          INTEGER REFERENCES partition(part_number) NOT
                          FOR REPLICATION,                          -- partition ID to which this purpose belongs
  bp_public             SMALLINT,                                   -- 1 = purpose is public, 0 = purpose is private
  bp_name               CHAR(32) NOT NULL,                          -- name of purpose
  site                  CHAR(32) NOT NULL,                          -- Site Name
  bp_checksum           INTEGER NOT NULL,                           -- Record checksum
  PRIMARY KEY           ( bp_name, site )
);
```

CREATE TABLE badgepurpose_save

```
(
  bp_guid               uniqueidentifier NOT NULL,                  -- Global unique record identifier
  bp_partition          INTEGER,                                    -- partition ID to which this purpose belongs
  bp_public             SMALLINT,                                   -- 1 = purpose is public, 0 = purpose is private
  bp_name               CHAR(32) NOT NULL,                          -- name of purpose
  site                  CHAR(32) NOT NULL,                          -- Site Name
  bp_checksum           INTEGER NOT NULL,                           -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                           -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,                  -- Global unique record identifier
  PRIMARY KEY           ( bp_guid, upd_mode desc, upd_guid )
);
```

CREATE TABLE badgesetup

```
(
  bc_printer            VARCHAR(500),                               -- Badging printer name
  bc_layout_path        CHAR(255),                                  -- badge layout file path
  bc_is_portrait        SMALLINT,                                   -- 1 = portrait is enabled
  bc_is_fingerprint     SMALLINT,                                   -- 1 = fingerprint is enabled
```

```
  bc_is_signature              SMALLINT,                                          -- 1 = signature is enabled
  bc_instance_portrait         INT,                                               -- number of portrait instances
  bc_instance_fingerprint      INT,                                               -- number of fingerprint instances
  bc_instance_signature        INT,                                               -- number of signature instances
  bc_device_portrait           VARCHAR(255),                                      -- device for portrait capture
  bc_device_fingerprint        VARCHAR(255),                                      -- device for fingerprint capture
  bc_device_signature          VARCHAR(255),                                      -- device for signature capture
  bc_guid                      uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  bc_checksum                  INTEGER NOT NULL,                                  -- Record checksum
  bc_station_id                INTEGER REFERENCES station(stn_id) NOT
                                 FOR REPLICATION NOT NULL,                        -- station ID
  bc_enforce_cropping_aspect   SMALLINT NOT NULL,                                 -- 1 = enforce aspect ration when cropping
  site                         CHAR(32) NOT NULL,                                 -- Site Name
  PRIMARY KEY                  ( site, bc_station_id)
);
```

CREATE TABLE badgesetup_save

```
(
  bc_printer                   VARCHAR(500),                                      -- Badging printer name
  bc_layout_path               CHAR(255),                                         -- badge layout file path
  bc_is_portrait               SMALLINT,
  bc_is_fingerprint            SMALLINT,
  bc_is_signature              SMALLINT,
  bc_instance_portrait         INT,
  bc_instance_fingerprint      INT,
  bc_instance_signature        INT,
  bc_device_portrait           VARCHAR(255),
  bc_device_fingerprint        VARCHAR(255),
  bc_device_signature          VARCHAR(255),
  site                         CHAR(32) NOT NULL,                                 -- Site Name
  bc_guid                      uniqueidentifier UNIQUE NOT NULL,                  -- Global unique record identifier
  bc_checksum                  INTEGER NOT NULL,                                  -- Record checksum
  upd_guid                     uniqueidentifier NOT NULL,                         -- Guid for update operation
  upd_mode                     SMALLINT NOT NULL,                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                DATETIME NOT NULL,                                 -- Date / Time of the update
  upd_checksum                 INTEGER NOT NULL,                                  -- Checksum
  upd_rowguid                  uniqueidentifier ROWGUIDCOL DEFAULT
                                 newid() UNIQUE NOT NULL,                         -- Global unique record identifier
  bc_station_id                INTEGER NOT NULL,                                  -- station ID
  bc_enforce_cropping_aspect   SMALLINT NOT NULL,                                 -- 1 = enforce aspect ration when cropping
  PRIMARY KEY                  (bc_guid, site, upd_guid)
);
```

CREATE TABLE badgesite

```
(
  bs_b_guid                    uniqueidentifier References badge(b_guid)
                                 NOT FOR REPLICATION,
  bs_internal                  SMALLINT NOT NULL,                                 -- 1 = internal record, not entered by operator
  bs_disabled                  SMALLINT NOT NULL,                                 -- 1 = disabled, 0 = active
  bs_event_priv                SMALLINT,                                          -- event privilege
  bs_exec_priv                 SMALLINT,                                          -- 1 = executive privilege
  bs_trace                     SMALLINT,                                          -- 1 = card trace on
  bs_override                  SMALLINT,                                          -- 1 = override on
  bs_dnld_sti_e                SMALLINT,                                          -- Download badge to STI-E if 1, otherwise 0
  bs_flag_a                    SMALLINT,                                          -- special flag A
  bs_flag_b                    SMALLINT,                                          -- special flag B
  bs_flag_c                    SMALLINT,                                          -- special flag C
  bs_block_xaction             SMALLINT,                                          -- Block Bages: 1 = block transaction, 0 = allow transaction
  site                         CHAR(32) NOT NULL,                                 -- Site Name
  bs_dnld_state                SMALLINT NOT NULL,                                 -- last download state (0 = never downloaded, 1 = active,
                                                                                    2 = disabled)
  bs_interlock1                INTEGER REFERENCES actioninterlock(ai_id)
```

```
                    NOT FOR REPLICATION,                    -- id of action interlock 1 to activate
bs_interlock1_value FLOAT,                                  -- value for action interlock 1
bs_interlock2       INTEGER REFERENCES actioninterlock(ai_id)
                    NOT FOR REPLICATION,                    -- id of action interlock 2 to activate
bs_interlock2_value FLOAT,                                  -- value for action interlock 2
bs_priority         SMALLINT,                               -- Badge Priority
bs_security_level   SMALLINT,                               -- security level 0-99
bs_access_grp_0     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 1
bs_access_grp_1     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 2
bs_access_grp_2     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 3
bs_access_grp_3     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 4
bs_access_grp_4     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 5
bs_access_grp_5     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 6
bs_access_grp_6     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 7
bs_access_grp_7     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 8
bs_access_tz_0      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,   -- access timezone # 1
bs_access_tz_1      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,   -- access timezone # 2
bs_access_tz_2      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,   -- access timezone # 3
bs_access_tz_3      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,   -- access timezone # 4
bs_access_tz_4      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,   -- access timezone # 5
bs_access_tz_5      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,   -- access timezone # 6
bs_access_tz_6      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,   -- access timezone # 7
bs_access_tz_7      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,   -- access timezone # 8
bs_ag_start0        DATETIME,                               -- start timestamp of access group 1
bs_ag_start1        DATETIME,                               -- start timestamp of access group 2
bs_ag_start2        DATETIME,                               -- start timestamp of access group 3
bs_ag_start3        DATETIME,                               -- start timestamp of access group 4
bs_ag_start4        DATETIME,                               -- start timestamp of access group 5
bs_ag_start5        DATETIME,                               -- start timestamp of access group 6
bs_ag_start6        DATETIME,                               -- start timestamp of access group 7
bs_ag_start7        DATETIME,                               -- start timestamp of access group 8
bs_ag_end0          DATETIME,                               -- end timestamp of access group 1
bs_ag_end1          DATETIME,                               -- end timestamp of access group 2
bs_ag_end2          DATETIME,                               -- end timestamp of access group 3
bs_ag_end3          DATETIME,                               -- end timestamp of access group 4
bs_ag_end4          DATETIME,                               -- end timestamp of access group 5
bs_ag_end5          DATETIME,                               -- end timestamp of access group 6
bs_ag_end6          DATETIME,                               -- end timestamp of access group 7
bs_ag_end7          DATETIME,                               -- end timestamp of access group 8
site                CHAR(32) NOT NULL,                      -- Site Name
bs_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
bs_checksum         INTEGER NOT NULL,                       -- Record checksum
bs_cardholder_active SMALLINT NOT NULL,                     -- same as csi_active from cardholder
                                                               site
bs_access_grp_8     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 9
bs_access_grp_9     INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 10
bs_access_grp_10    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 11
bs_access_grp_11    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 12
bs_access_grp_12    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 13
bs_access_grp_13    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 14
bs_access_grp_14    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 15
bs_access_grp_15    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 16
bs_access_grp_16    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 17
bs_access_grp_17    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 18
bs_access_grp_18    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 19
bs_access_grp_19    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 20
bs_access_grp_20    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 21
bs_access_grp_21    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 22
bs_access_grp_22    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 23
bs_access_grp_23    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 24
bs_access_grp_24    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 25
bs_access_grp_25    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 26
bs_access_grp_26    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 27
bs_access_grp_27    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 28
bs_access_grp_28    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,   -- access group # 29
```

```
bs_access_grp_29    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 30
bs_access_grp_30    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 31
bs_access_grp_31    INTEGER NULL REFERENCES accgroup(ag_id) NOT FOR REPLICATION,    -- access group # 32
bs_access_tz_8      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 1
bs_access_tz_9      INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 2
bs_access_tz_10     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 3
bs_access_tz_11     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 4
bs_access_tz_12     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 5
bs_access_tz_13     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 6
bs_access_tz_14     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 7
bs_access_tz_15     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 8
bs_access_tz_16     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 1
bs_access_tz_17     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 2
bs_access_tz_18     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 3
bs_access_tz_19     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 4
bs_access_tz_20     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 5
bs_access_tz_21     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 6
bs_access_tz_22     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 7
bs_access_tz_23     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 8
bs_access_tz_24     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 1
bs_access_tz_25     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 2
bs_access_tz_26     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 3
bs_access_tz_27     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 4
bs_access_tz_28     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 5
bs_access_tz_29     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 6
bs_access_tz_30     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 7
bs_access_tz_31     INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,    -- access timezone # 8
bs_ag_start8        DATETIME,                                                       -- start timestamp of access group 1
bs_ag_start9        DATETIME,                                                       -- start timestamp of access group 2
bs_ag_start10       DATETIME,                                                       -- start timestamp of access group 3
bs_ag_start11       DATETIME,                                                       -- start timestamp of access group 4
bs_ag_start12       DATETIME,                                                       -- start timestamp of access group 5
bs_ag_start13       DATETIME,                                                       -- start timestamp of access group 6
bs_ag_start14       DATETIME,                                                       -- start timestamp of access group 7
bs_ag_start15       DATETIME,                                                       -- start timestamp of access group 8
bs_ag_start16       DATETIME,                                                       -- start timestamp of access group 1
bs_ag_start17       DATETIME,                                                       -- start timestamp of access group 2
bs_ag_start18       DATETIME,                                                       -- start timestamp of access group 3
bs_ag_start19       DATETIME,                                                       -- start timestamp of access group 4
bs_ag_start20       DATETIME,                                                       -- start timestamp of access group 5
bs_ag_start21       DATETIME,                                                       -- start timestamp of access group 6
bs_ag_start22       DATETIME,                                                       -- start timestamp of access group 7
bs_ag_start23       DATETIME,                                                       -- start timestamp of access group 8
bs_ag_start24       DATETIME,                                                       -- start timestamp of access group 1
bs_ag_start25       DATETIME,                                                       -- start timestamp of access group 2
bs_ag_start26       DATETIME,                                                       -- start timestamp of access group 3
bs_ag_start27       DATETIME,                                                       -- start timestamp of access group 4
bs_ag_start28       DATETIME,                                                       -- start timestamp of access group 5
bs_ag_start29       DATETIME,                                                       -- start timestamp of access group 6
bs_ag_start30       DATETIME,                                                       -- start timestamp of access group 7
bs_ag_start31       DATETIME,                                                       -- start timestamp of access group 8
bs_ag_end8          DATETIME,                                                       -- end timestamp of access group 9
bs_ag_end9          DATETIME,                                                       -- end timestamp of access group 10
bs_ag_end10         DATETIME,                                                       -- end timestamp of access group 11
bs_ag_end11         DATETIME,                                                       -- end timestamp of access group 12
bs_ag_end12         DATETIME,                                                       -- end timestamp of access group 13
bs_ag_end13         DATETIME,                                                       -- end timestamp of access group 14
bs_ag_end14         DATETIME,                                                       -- end timestamp of access group 15
bs_ag_end15         DATETIME,                                                       -- end timestamp of access group 16
bs_ag_end16         DATETIME,                                                       -- end timestamp of access group 17
bs_ag_end17         DATETIME,                                                       -- end timestamp of access group 18
bs_ag_end18         DATETIME,                                                       -- end timestamp of access group 19
bs_ag_end19         DATETIME,                                                       -- end timestamp of access group 20
bs_ag_end20         DATETIME,                                                       -- end timestamp of access group 21
bs_ag_end21         DATETIME,                                                       -- end timestamp of access group 22
```

```
  bs_ag_end22           DATETIME,                                          -- end timestamp of access group 23
  bs_ag_end23           DATETIME,                                          -- end timestamp of access group 24
  bs_ag_end24           DATETIME,                                          -- end timestamp of access group 25
  bs_ag_end25           DATETIME,                                          -- end timestamp of access group 26
  bs_ag_end26           DATETIME,                                          -- end timestamp of access group 27
  bs_ag_end27           DATETIME,                                          -- end timestamp of access group 28
  bs_ag_end28           DATETIME,                                          -- end timestamp of access group 29
  bs_ag_end29           DATETIME,                                          -- end timestamp of access group 30
  bs_ag_end30           DATETIME,                                          -- end timestamp of access group 31
  bs_ag_end31           DATETIME,                                          -- end timestamp of access group 32
  bs_default_floor      SMALLINT,                                          -- Otis Compass default floor for this
                                                                              badge
  bs_otis_pin_floor_mask INTEGER REFERENCES floormask(fm_id)  NOT FOR REPLICATION,  -- id of floor mask used with Otis
                                                                              Compass pin code is received
  PRIMARY KEY           NONCLUSTERED (bs_b_guid, site)
);
  CREATE INDEX          bs_cardholder_active ON badgesite ( bs_cardholder_active );
  CREATE INDEX          bs_dnld_state ON badgesite ( bs_dnld_state );
  CREATE INDEX          bs_b_guid ON badgesite ( bs_b_guid );
```

CREATE TABLE badgesite_save

```
(
  bs_b_guid             uniqueidentifier,
  bs_internal           smallint NOT NULL,
  bs_disabled           SMALLINT NOT NULL,                 -- 1 = disabled, 0 = active
  bs_event_priv         SMALLINT,                          -- event privilege
  bs_exec_priv          SMALLINT,                          -- 1 = executive privilege
  bs_trace              SMALLINT,                          -- 1 = card trace on
  bs_override           SMALLINT,                          -- 1 = override on
  bs_dnld_sti_e         SMALLINT,                          -- Download badge to STI-E if 1, otherwise 0
  bs_flag_a             SMALLINT,                          -- special flag A
  bs_flag_b             SMALLINT,                          -- special flag B
  bs_flag_c             SMALLINT,                          -- special flag C
  bs_block_xaction      SMALLINT,                          -- Block Bages: 1 = block transaction, 0 = allow transaction
  site                  CHAR(32) NOT NULL,                 -- Site Name
  bs_dnld_state         SMALLINT NOT NULL,                 -- last download state (0 = never downloaded, 1 = active,
                                                              2 = disabled)
  bs_interlock1         INTEGER,                           -- id of action interlock 1 to activate
  bs_interlock1_value   FLOAT,                             -- value for action interlock 1
  bs_interlock2         INTEGER,                           -- id of action interlock 2 to activate
  bs_interlock2_value   FLOAT,                             -- value for action interlock 2
  bs_priority           SMALLINT,                          -- Badge Priority
  bs_security_level     SMALLINT,                          -- security level 0-99
  bs_access_grp_0       INTEGER NULL,                      -- access group # 1
  bs_access_grp_1       INTEGER NULL,                      -- access group # 2
  bs_access_grp_2       INTEGER NULL,                      -- access group # 3
  bs_access_grp_3       INTEGER NULL,                      -- access group # 4
  bs_access_grp_4       INTEGER NULL,                      -- access group # 5
  bs_access_grp_5       INTEGER NULL,                      -- access group # 6
  bs_access_grp_6       INTEGER NULL,                      -- access group # 7
  bs_access_grp_7       INTEGER NULL,                      -- access group # 8
  bs_access_tz_0        INTEGER NULL,                      -- access timezone # 1
  bs_access_tz_1        INTEGER NULL,                      -- access timezone # 2
  bs_access_tz_2        INTEGER NULL,                      -- access timezone # 3
  bs_access_tz_3        INTEGER NULL,                      -- access timezone # 4
  bs_access_tz_4        INTEGER NULL,                      -- access timezone # 5
  bs_access_tz_5        INTEGER NULL,                      -- access timezone # 6
  bs_access_tz_6        INTEGER NULL,                      -- access timezone # 7
  bs_access_tz_7        INTEGER NULL,                      -- access timezone # 8
  bs_ag_start0          DATETIME,                          -- start timestamp of access group 1
  bs_ag_start1          DATETIME,                          -- start timestamp of access group 2
  bs_ag_start2          DATETIME,                          -- start timestamp of access group 3
  bs_ag_start3          DATETIME,                          -- start timestamp of access group 4
```

```
bs_ag_start4            DATETIME,                                   -- start timestamp of access group 5
bs_ag_start5            DATETIME,                                   -- start timestamp of access group 6
bs_ag_start6            DATETIME,                                   -- start timestamp of access group 7
bs_ag_start7            DATETIME,                                   -- start timestamp of access group 8
bs_ag_end0              DATETIME,                                   -- end timestamp of access group 1
bs_ag_end1              DATETIME,                                   -- end timestamp of access group 2
bs_ag_end2              DATETIME,                                   -- end timestamp of access group 3
bs_ag_end3              DATETIME,                                   -- end timestamp of access group 4
bs_ag_end4              DATETIME,                                   -- end timestamp of access group 5
bs_ag_end5              DATETIME,                                   -- end timestamp of access group 6
bs_ag_end6              DATETIME,                                   -- end timestamp of access group 7
bs_ag_end7              DATETIME,                                   -- end timestamp of access group 8
site                    CHAR(32) NOT NULL,                          -- Site Name
bs_guid                 uniqueidentifier NOT NULL,                  -- Global unique record identifier
bs_checksum             INTEGER NOT NULL,                           -- Record checksum
upd_guid                uniqueidentifier NOT NULL,                  -- Guid for update operation
upd_mode                SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
upd_timestamp           DATETIME NOT NULL,                          -- Date / Time of the update
upd_checksum            INTEGER NOT NULL,                           -- Checksum
upd_rowguid             uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,                  -- Global unique record identifier
bs_cardholder_active    SMALLINT NOT NULL,                          -- same as csi_active from cardholder table
bs_access_grp_8         INTEGER NULL,                               -- access group # 9
bs_access_grp_9         INTEGER NULL,                               -- access group # 10
bs_access_grp_10        INTEGER NULL,                               -- access group # 11
bs_access_grp_11        INTEGER NULL,                               -- access group # 12
bs_access_grp_12        INTEGER NULL,                               -- access group # 13
bs_access_grp_13        INTEGER NULL,                               -- access group # 14
bs_access_grp_14        INTEGER NULL,                               -- access group # 15
bs_access_grp_15        INTEGER NULL,                               -- access group # 16
bs_access_grp_16        INTEGER NULL,                               -- access group # 17
bs_access_grp_17        INTEGER NULL,                               -- access group # 18
bs_access_grp_18        INTEGER NULL,                               -- access group # 19
bs_access_grp_19        INTEGER NULL,                               -- access group # 20
bs_access_grp_20        INTEGER NULL,                               -- access group # 21
bs_access_grp_21        INTEGER NULL,                               -- access group # 22
bs_access_grp_22        INTEGER NULL,                               -- access group # 23
bs_access_grp_23        INTEGER NULL,                               -- access group # 24
bs_access_grp_24        INTEGER NULL,                               -- access group # 25
bs_access_grp_25        INTEGER NULL,                               -- access group # 26
bs_access_grp_26        INTEGER NULL,                               -- access group # 27
bs_access_grp_27        INTEGER NULL,                               -- access group # 28
bs_access_grp_28        INTEGER NULL,                               -- access group # 29
bs_access_grp_29        INTEGER NULL,                               -- access group # 30
bs_access_grp_30        INTEGER NULL,                               -- access group # 31
bs_access_grp_31        INTEGER NULL,                               -- access group # 32
bs_access_tz_8          INTEGER NULL,                               -- access timezone # 1
bs_access_tz_9          INTEGER NULL,                               -- access timezone # 2
bs_access_tz_10         INTEGER NULL,                               -- access timezone # 3
bs_access_tz_11         INTEGER NULL,                               -- access timezone # 4
bs_access_tz_12         INTEGER NULL,                               -- access timezone # 5
bs_access_tz_13         INTEGER NULL,                               -- access timezone # 6
bs_access_tz_14         INTEGER NULL,                               -- access timezone # 7
bs_access_tz_15         INTEGER NULL,                               -- access timezone # 8
bs_access_tz_16         INTEGER NULL,                               -- access timezone # 1
bs_access_tz_17         INTEGER NULL,                               -- access timezone # 2
bs_access_tz_18         INTEGER NULL,                               -- access timezone # 3
bs_access_tz_19         INTEGER NULL,                               -- access timezone # 4
bs_access_tz_20         INTEGER NULL,                               -- access timezone # 5
bs_access_tz_21         INTEGER NULL,                               -- access timezone # 6
bs_access_tz_22         INTEGER NULL,                               -- access timezone # 7
bs_access_tz_23         INTEGER NULL,                               -- access timezone # 8
bs_access_tz_24         INTEGER NULL,                               -- access timezone # 1
bs_access_tz_25         INTEGER NULL,                               -- access timezone # 2
bs_access_tz_26         INTEGER NULL,                               -- access timezone # 3
```

```
  bs_access_tz_27       INTEGER NULL,                                   -- access timezone # 4
  bs_access_tz_28       INTEGER NULL,                                   -- access timezone # 5
  bs_access_tz_29       INTEGER NULL,                                   -- access timezone # 6
  bs_access_tz_30       INTEGER NULL,                                   -- access timezone # 7
  bs_access_tz_31       INTEGER NULL,                                   -- access timezone # 8
  bs_ag_start8          DATETIME,                                       -- start timestamp of access group 1
  bs_ag_start9          DATETIME,                                       -- start timestamp of access group 2
  bs_ag_start10         DATETIME,                                       -- start timestamp of access group 3
  bs_ag_start11         DATETIME,                                       -- start timestamp of access group 4
  bs_ag_start12         DATETIME,                                       -- start timestamp of access group 5
  bs_ag_start13         DATETIME,                                       -- start timestamp of access group 6
  bs_ag_start14         DATETIME,                                       -- start timestamp of access group 7
  bs_ag_start15         DATETIME,                                       -- start timestamp of access group 8
  bs_ag_start16         DATETIME,                                       -- start timestamp of access group 1
  bs_ag_start17         DATETIME,                                       -- start timestamp of access group 2
  bs_ag_start18         DATETIME,                                       -- start timestamp of access group 3
  bs_ag_start19         DATETIME,                                       -- start timestamp of access group 4
  bs_ag_start20         DATETIME,                                       -- start timestamp of access group 5
  bs_ag_start21         DATETIME,                                       -- start timestamp of access group 6
  bs_ag_start22         DATETIME,                                       -- start timestamp of access group 7
  bs_ag_start23         DATETIME,                                       -- start timestamp of access group 8
  bs_ag_start24         DATETIME,                                       -- start timestamp of access group 1
  bs_ag_start25         DATETIME,                                       -- start timestamp of access group 2
  bs_ag_start26         DATETIME,                                       -- start timestamp of access group 3
  bs_ag_start27         DATETIME,                                       -- start timestamp of access group 4
  bs_ag_start28         DATETIME,                                       -- start timestamp of access group 5
  bs_ag_start29         DATETIME,                                       -- start timestamp of access group 6
  bs_ag_start30         DATETIME,                                       -- start timestamp of access group 7
  bs_ag_start31         DATETIME,                                       -- start timestamp of access group 8
  bs_ag_end8            DATETIME,                                       -- end timestamp of access group 9
  bs_ag_end9            DATETIME,                                       -- end timestamp of access group 10
  bs_ag_end10           DATETIME,                                       -- end timestamp of access group 11
  bs_ag_end11           DATETIME,                                       -- end timestamp of access group 12
  bs_ag_end12           DATETIME,                                       -- end timestamp of access group 13
  bs_ag_end13           DATETIME,                                       -- end timestamp of access group 14
  bs_ag_end14           DATETIME,                                       -- end timestamp of access group 15
  bs_ag_end15           DATETIME,                                       -- end timestamp of access group 16
  bs_ag_end16           DATETIME,                                       -- end timestamp of access group 17
  bs_ag_end17           DATETIME,                                       -- end timestamp of access group 18
  bs_ag_end18           DATETIME,                                       -- end timestamp of access group 19
  bs_ag_end19           DATETIME,                                       -- end timestamp of access group 20
  bs_ag_end20           DATETIME,                                       -- end timestamp of access group 21
  bs_ag_end21           DATETIME,                                       -- end timestamp of access group 22
  bs_ag_end22           DATETIME,                                       -- end timestamp of access group 23
  bs_ag_end23           DATETIME,                                       -- end timestamp of access group 24
  bs_ag_end24           DATETIME,                                       -- end timestamp of access group 25
  bs_ag_end25           DATETIME,                                       -- end timestamp of access group 26
  bs_ag_end26           DATETIME,                                       -- end timestamp of access group 27
  bs_ag_end27           DATETIME,                                       -- end timestamp of access group 28
  bs_ag_end28           DATETIME,                                       -- end timestamp of access group 29
  bs_ag_end29           DATETIME,                                       -- end timestamp of access group 30
  bs_ag_end30           DATETIME,                                       -- end timestamp of access group 31
  bs_ag_end31           DATETIME,                                       -- end timestamp of access group 32
  bs_default_floor      SMALLINT,                                       -- Otis Compass default floor for this badge
  bs_otis_pin_floor_mask INTEGER NULL,                                  -- id of floor mask used with Otis Compass pin code is received
  PRIMARY KEY           NONCLUSTERED (bs_b_guid, site, upd_mode desc, upd_guid)
);
```

CREATE TABLE badgestatus

```
(
  bs_number_str         VARCHAR(100) REFERENCES badge(b_number_str)
                          NOT FOR REPLICATION,                          -- badge number
  bs_valid_hist_type    INTEGER,                                        -- last valid badging xaction type
```

```
  bs_invalid_hist_type      INTEGER,                                        -- last invalid badging xaction type
  bs_valid_timestamp        DATETIME,                                       -- last valid badging time
  bs_invalid_timestamp      DATETIME,                                       -- last invalid badging time
  bs_valid_term_id          INTEGER,                                        -- last valid badging terminal ID
  bs_invalid_term_id        INTEGER,                                        -- last invalid badging terminal ID
  bs_inxit_status           SMALLINT,                                       -- current IN/OUT status
  bs_pin_retry_cnt          SMALLINT,                                       -- number of pin code retries (central mode)
  bs_apb_timer              INTEGER,                                        -- anti-passback timer (central mode)
  site                      CHAR(32) NOT NULL,                              -- Site Name
  bs_guid                   uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
  bs_checksum               INTEGER NOT NULL,                               -- Record checksum
  PRIMARY KEY               (bs_number_str, site)
);
  CREATE INDEX              bs_valid_timestamp ON badgestatus ( bs_valid_timestamp );
```

CREATE TABLE [dbo].BadgeStatusSave

Table allows restoring personnel to last badge location after a muster. Only personnel inside the zone at muster start are saved here.

```
  bss_zone_id               INTEGER REFERENCES MusterDefinition(m_id)
                              NOT FOR REPLICATION,                          -- Zone to which this data belongs
  bss_number_str            VARCHAR(100) NOT NULL,                          -- badge number
  bss_valid_hist_type       INTEGER,                                        -- last valid badging xaction type
  bss_invalid_hist_type     INTEGER,                                        -- last invalid badging xaction type
  bss_valid_timestamp       DATETIME,                                       -- last valid badging time
  bss_invalid_timestamp     DATETIME,                                       -- last invalid badging time
  bss_valid_term_id         INTEGER,                                        -- last valid badging terminal ID
  bss_invalid_term_id       INTEGER,                                        -- last invalid badging terminal ID
  bss_inxit_status          SMALLINT,                                       -- current IN/OUT status
  bss_pin_retry_cnt         SMALLINT,                                       -- number of pin code retries (central mode)
  bss_apb_timer             INTEGER,                                        -- anti-passback timer (central mode)
  site                      CHAR(32) NOT NULL,                              -- Site Name
  bss_guid                  uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
  bss_checksum              INTEGER NOT NULL,                               -- Record checksum
  PRIMARY KEY               (bss_number_str, bss_zone_id),
);
```

CREATE TABLE badge_save

```
(
  b_partition               INTEGER NOT NULL,                               -- partition ID to which this badge belongs
  b_public                  SMALLINT,                                       -- 1 = badge is public, 0 = badge is private
  b_number_str              VARCHAR(100) NOT NULL,                          -- badge number
  b_cardholder_id           INTEGER,                                        -- cardholder ID of cardholder who has this badge
  b_alpha                   CHAR(4) NULL,                                   -- alpha field of badge
  b_issue                   SMALLINT,                                       -- issue level
  b_description             CHAR(64) NULL,                                  -- user entered description
  b_disabled                SMALLINT NOT NULL,                              -- 1 = disabled, 0 = active
  b_exp_timestamp           DATETIME,                                       -- expiration timestamp
  b_start_timestamp         DATETIME,                                       -- start timestamp
  b_pin                     INTEGER,                                        -- pin code
  b_event_priv              SMALLINT,                                       -- event privilege
  b_exec_priv               SMALLINT,                                       -- 1 = executive privilege
  b_trace                   SMALLINT,                                       -- 1 = card trace on
  b_override                SMALLINT,                                       -- 1 = override on
  b_access_grp_0            INTEGER NULL,                                   -- access group # 1
  b_access_grp_1            INTEGER NULL,                                   -- access group # 2
  b_access_grp_2            INTEGER NULL,                                   -- access group # 3
  b_access_grp_3            INTEGER NULL,                                   -- access group # 4
  b_access_grp_4            INTEGER NULL,                                   -- access group # 5
  b_access_grp_5            INTEGER NULL,                                   -- access group # 6
  b_access_grp_6            INTEGER NULL,                                   -- access group # 7
  b_access_grp_7            INTEGER NULL,                                   -- access group # 8
  b_access_tz_0             INTEGER NULL,                                   -- access timezone # 1
```

```
b_access_tz_1           INTEGER NULL,                          -- access timezone # 2
b_access_tz_2           INTEGER NULL,                          -- access timezone # 3
b_access_tz_3           INTEGER NULL,                          -- access timezone # 4
b_access_tz_4           INTEGER NULL,                          -- access timezone # 5
b_access_tz_5           INTEGER NULL,                          -- access timezone # 6
b_access_tz_6           INTEGER NULL,                          -- access timezone # 7
b_access_tz_7           INTEGER NULL,                          -- access timezone # 8
b_reason                CHAR(32) NULL,                         -- reason for badge issue
b_design                INTEGER NULL,                          -- badge design ID
b_interlock1            INTEGER,                               -- id of action interlock 1 to activate
b_interlock1_value      FLOAT,                                 -- value for action interlock 1
b_interlock2            INTEGER,                               -- id of action interlock 2 to activate
b_interlock2_value      FLOAT,                                 -- value for action interlock 2
b_priority              SMALLINT,                              -- Badge Priority
b_dnld_sti_e            SMALLINT,                              -- Download badge to STI-E if 1, otherwise 0
b_ag_start0             DATETIME,                              -- start timestamp of access group 1
b_ag_start1             DATETIME,                              -- start timestamp of access group 2
b_ag_start2             DATETIME,                              -- start timestamp of access group 3
b_ag_start3             DATETIME,                              -- start timestamp of access group 4
b_ag_start4             DATETIME,                              -- start timestamp of access group 5
b_ag_start5             DATETIME,                              -- start timestamp of access group 6
b_ag_start6             DATETIME,                              -- start timestamp of access group 7
b_ag_start7             DATETIME,                              -- start timestamp of access group 8
b_ag_end0               DATETIME,                              -- end timestamp of access group 1
b_ag_end1               DATETIME,                              -- end timestamp of access group 2
b_ag_end2               DATETIME,                              -- end timestamp of access group 3
b_ag_end3               DATETIME,                              -- end timestamp of access group 4
b_ag_end4               DATETIME,                              -- end timestamp of access group 5
b_ag_end5               DATETIME,                              -- end timestamp of access group 6
b_ag_end6               DATETIME,                              -- end timestamp of access group 7
b_ag_end7               DATETIME,                              -- end timestamp of access group 8
b_facility_code         INTEGER,                               -- facility code 0 to 65535
b_security_level        SMALLINT,                              -- security level 0-99
b_flag_a                SMALLINT,                              -- special flag A
b_flag_b                SMALLINT,                              -- special flag B
b_flag_c                SMALLINT,                              -- special flag C
b_block_xaction         SMALLINT,                              -- Block Bages: 1 = block transaction, 0 = allow transaction
site                    CHAR(32) NOT NULL,                     -- Site Name
site                    CHAR(32) NOT NULL,                     -- Site Name
b_guid                  uniqueidentifier NOT NULL,             -- Global unique record identifier
b_checksum              INTEGER NOT NULL,                      -- Record checksum
upd_guid                uniqueidentifier  NOT NULL,            -- Guid for update operation
upd_mode                SMALLINT NOT NULL,                     -- Update mode (0 - delete, 1 - edit, 2 - new)
upd_timestamp           DATETIME NOT NULL,                     -- Date / Time of the update
upd_checksum            INTEGER NOT NULL,                      -- Checksum
upd_rowguid             uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,             -- Global unique record identifier
b_access_grp_8          INTEGER NULL,                          -- access group # 9
b_access_grp_9          INTEGER NULL,                          -- access group # 10
b_access_grp_10         INTEGER NULL,                          -- access group # 11
b_access_grp_11         INTEGER NULL,                          -- access group # 12
b_access_grp_12         INTEGER NULL,                          -- access group # 13
b_access_grp_13         INTEGER NULL,                          -- access group # 14
b_access_grp_14         INTEGER NULL,                          -- access group # 15
b_access_grp_15         INTEGER NULL,                          -- access group # 16
b_access_grp_16         INTEGER NULL,                          -- access group # 17
b_access_grp_17         INTEGER NULL,                          -- access group # 18
b_access_grp_18         INTEGER NULL,                          -- access group # 19
b_access_grp_19         INTEGER NULL,                          -- access group # 20
b_access_grp_20         INTEGER NULL,                          -- access group # 21
b_access_grp_21         INTEGER NULL,                          -- access group # 22
b_access_grp_22         INTEGER NULL,                          -- access group # 23
b_access_grp_23         INTEGER NULL,                          -- access group # 24
b_access_grp_24         INTEGER NULL,                          -- access group # 25
b_access_grp_25         INTEGER NULL,                          -- access group # 26
```

```
b_access_grp_26     INTEGER NULL,     -- access group # 27
b_access_grp_27     INTEGER NULL,     -- access group # 28
b_access_grp_28     INTEGER NULL,     -- access group # 29
b_access_grp_29     INTEGER NULL,     -- access group # 30
b_access_grp_30     INTEGER NULL,     -- access group # 31
b_access_grp_31     INTEGER NULL,     -- access group # 32
b_access_tz_8       INTEGER NULL,     -- access timezone # 9
b_access_tz_9       INTEGER NULL,     -- access timezone # 10
b_access_tz_10      INTEGER NULL,     -- access timezone # 11
b_access_tz_11      INTEGER NULL,     -- access timezone # 12
b_access_tz_12      INTEGER NULL,     -- access timezone # 13
b_access_tz_13      INTEGER NULL,     -- access timezone # 14
b_access_tz_14      INTEGER NULL,     -- access timezone # 15
b_access_tz_15      INTEGER NULL,     -- access timezone # 16
b_access_tz_16      INTEGER NULL,     -- access timezone # 17
b_access_tz_17      INTEGER NULL,     -- access timezone # 18
b_access_tz_18      INTEGER NULL,     -- access timezone # 19
b_access_tz_19      INTEGER NULL,     -- access timezone # 20
b_access_tz_20      INTEGER NULL,     -- access timezone # 21
b_access_tz_21      INTEGER NULL,     -- access timezone # 22
b_access_tz_22      INTEGER NULL,     -- access timezone # 23
b_access_tz_23      INTEGER NULL,     -- access timezone # 24
b_access_tz_24      INTEGER NULL,     -- access timezone # 25
b_access_tz_25      INTEGER NULL,     -- access timezone # 26
b_access_tz_26      INTEGER NULL,     -- access timezone # 27
b_access_tz_27      INTEGER NULL,     -- access timezone # 28
b_access_tz_28      INTEGER NULL,     -- access timezone # 29
b_access_tz_29      INTEGER NULL,     -- access timezone # 30
b_access_tz_30      INTEGER NULL,     -- access timezone # 31
b_access_tz_31      INTEGER NULL,     -- access timezone # 32
b_ag_start8         DATETIME,         -- start timestamp of access group 9
b_ag_start9         DATETIME,         -- start timestamp of access group 10
b_ag_start10        DATETIME,         -- start timestamp of access group 11
b_ag_start11        DATETIME,         -- start timestamp of access group 12
b_ag_start12        DATETIME,         -- start timestamp of access group 13
b_ag_start13        DATETIME,         -- start timestamp of access group 14
b_ag_start14        DATETIME,         -- start timestamp of access group 15
b_ag_start15        DATETIME,         -- start timestamp of access group 16
b_ag_start16        DATETIME,         -- start timestamp of access group 17
b_ag_start17        DATETIME,         -- start timestamp of access group 18
b_ag_start18        DATETIME,         -- start timestamp of access group 19
b_ag_start19        DATETIME,         -- start timestamp of access group 20
b_ag_start20        DATETIME,         -- start timestamp of access group 21
b_ag_start21        DATETIME,         -- start timestamp of access group 22
b_ag_start22        DATETIME,         -- start timestamp of access group 23
b_ag_start23        DATETIME,         -- start timestamp of access group 24
b_ag_start24        DATETIME,         -- start timestamp of access group 25
b_ag_start25        DATETIME,         -- start timestamp of access group 26
b_ag_start26        DATETIME,         -- start timestamp of access group 27
b_ag_start27        DATETIME,         -- start timestamp of access group 28
b_ag_start28        DATETIME,         -- start timestamp of access group 29
b_ag_start29        DATETIME,         -- start timestamp of access group 30
b_ag_start30        DATETIME,         -- start timestamp of access group 31
b_ag_start31        DATETIME,         -- start timestamp of access group 32
b_ag_end8           DATETIME,         -- end timestamp of access group 9
b_ag_end9           DATETIME,         -- end timestamp of access group 10
b_ag_end10          DATETIME,         -- end timestamp of access group 11
b_ag_end11          DATETIME,         -- end timestamp of access group 12
b_ag_end12          DATETIME,         -- end timestamp of access group 13
b_ag_end13          DATETIME,         -- end timestamp of access group 14
b_ag_end14          DATETIME,         -- end timestamp of access group 15
b_ag_end15          DATETIME,         -- end timestamp of access group 16
b_ag_end16          DATETIME,         -- end timestamp of access group 17
b_ag_end17          DATETIME,         -- end timestamp of access group 18
b_ag_end18          DATETIME,         -- end timestamp of access group 19
```

```
  b_ag_end19            DATETIME,                                   -- end timestamp of access group 20
  b_ag_end20            DATETIME,                                   -- end timestamp of access group 21
  b_ag_end21            DATETIME,                                   -- end timestamp of access group 22
  b_ag_end22            DATETIME,                                   -- end timestamp of access group 23
  b_ag_end23            DATETIME,                                   -- end timestamp of access group 24
  b_ag_end24            DATETIME,                                   -- end timestamp of access group 25
  b_ag_end25            DATETIME,                                   -- end timestamp of access group 26
  b_ag_end26            DATETIME,                                   -- end timestamp of access group 27
  b_ag_end27            DATETIME,                                   -- end timestamp of access group 28
  b_ag_end28            DATETIME,                                   -- end timestamp of access group 29
  b_ag_end29            DATETIME,                                   -- end timestamp of access group 30
  b_ag_end30            DATETIME,                                   -- end timestamp of access group 31
  b_ag_end31            DATETIME,                                   -- end timestamp of access group 32
  b_purpose             uniqueidentifier NULL,
  b_data_style          SMALLINT NOT NULL,                          -- 0 = normal, 1 = FASCN
  b_default_floor       SMALLINT,                                   -- Otis Compass default floor for this badge
  b_otis_pin_floor_mask INTEGER NULL,                               -- id of floor mask used with Otis Compass pin code is received
  b_operator            CHAR(32),                                   -- Operator that last modified record
  b_edit_time           DATETIME,                                   -- Time this record was last edited
  b_technology          INTEGER NOT NULL,                           -- technology of badge format
  b_bits                INTEGER NOT NULL,                           -- the number of bits in the badge format
  b_qualifier           INTEGER NOT NULL,                           -- format qualifier for the badge format
  PRIMARY KEY           (b_number_str, upd_mode desc, upd_guid)
);
```

CREATE TABLE cardholder

```
(
  c_partition           INTEGER REFERENCES partition(part_number)
                          NOT FOR REPLICATION,                      -- partition ID to which this cardholder belongs
  c_public              SMALLINT,                                   -- 1 = cardholder is public, 0 = cardholder is private
  c_id                  INTEGER IDENTITY NOT FOR REPLICATION UNIQUE, -- unique database ID
  c_lname               CHAR(25) NOT NULL,                          -- last name
  c_mname               CHAR(25) NULL,                              -- middle name
  c_fname               CHAR(25) NOT NULL,                          -- first name
  c_nick_name           CHAR(25) NULL,                              -- ID, must be unique if not 0
  c_addr                CHAR(64) NULL,                              -- street address
  c_addr1               CHAR(32) NULL,                              -- city
  c_addr2               CHAR(32) NULL,                              -- state
  c_addr3               CHAR(32) NULL,                              -- zip code
  c_phone               CHAR(16) NULL,                              -- phone number
  c_ext                 CHAR(6) NULL,                               -- phone extension
  c_s_timestamp         DATETIME,                                   -- start timestamp
  c_t_timestamp         DATETIME,                                   -- terminate timestamp
  c_card_type           SMALLINT,                                   -- 0 = regular, 1 = visitor
  c_dept_id             INTEGER NULL REFERENCES dept(dept_id)
                          NOT FOR REPLICATION,                      -- database ID of department
  c_company_id          INTEGER NULL REFERENCES company(company_id)
                          NOT FOR REPLICATION,                      -- database ID of company
  c_sponsor_id          INTEGER NULL REFERENCES cardholder(c_id)
                          NOT FOR REPLICATION,                      -- database ID of sponsor (visitor only)
  c_guard               SMALLINT,                                   -- Can be guard or not
  c_suite               CHAR(32) NULL,                              -- suit information
  site                  CHAR(32) NOT NULL,                          -- Site Name
  c_guid                uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  c_checksum            INTEGER NOT NULL,                           -- Record checksum
  c_email               CHAR(64) NULL,                              -- email address
  c_operator            CHAR(32),                                   -- Operator that last modified record
  c_edit_time           DATETIME,                                   -- Time this record was last edited
  PRIMARY KEY           ( c_partition, c_id )
);
  CREATE INDEX          c_public ON cardholder ( c_public );
  CREATE INDEX          c_lname ON cardholder ( c_lname );
  CREATE INDEX          c_fname ON cardholder ( c_fname );
```

```
CREATE INDEX        c_mname ON cardholder ( c_mname );
CREATE INDEX        c_card_type ON cardholder ( c_card_type );
CREATE INDEX        c_begin ON cardholder ( c_s_timestamp );
CREATE INDEX        c_end ON cardholder ( c_t_timestamp );
CREATE INDEX        c_nick_name ON cardholder ( c_nick_name );
CREATE INDEX        c_guard ON cardholder ( c_guard );
CREATE INDEX        c_company_id ON cardholder ( c_company_id );
```

CREATE TABLE cardevent

```
(
  ce_id             INTEGER IDENTITY NOT FOR REPLICATION
                      PRIMARY KEY,                          -- unique database ID
  ce_panel_id       INTEGER REFERENCES panel(p_panel_id)
                      NOT FOR REPLICATION,                  -- ID of panel this card event pertains to
  ce_event_name     CHAR(32),                               -- name of this event
  ce_event_number   SMALLINT,                               -- unique event number for the selected panel
  ce_trigger_type   SMALLINT,                               -- 0 = "Card Only", 1 = "Card/PIN Code", 2 = "Card/Keypad
                                                              Code", 3 = "Card/PIN/Keypad Code", 4 = "Any Void Card"
  ce_priv_level     SMALLINT,                               -- priviledge level required for this event
  ce_access_code    CHAR(4) NULL,                           -- keypad code to trigger event
  ce_event_period   SMALLINT,                               -- event duration in minutes
  ce_output_group   INTEGER NULL REFERENCES outputgrp(op_group_id)
                      NOT FOR REPLICATION,                  -- ID of output group that can be activated
  ce_input_group    INTEGER NULL REFERENCES inputgrp(ip_group_id)
                      NOT FOR REPLICATION,                  -- ID of input group that can be suppressed
  ce_ip_suppress    SMALLINT,                               -- 0 = don't suppress input group, 1 = suppress input group
  ce_door_strike    SMALLINT,                               -- 0 = don't operate door strike, 1 = operate door strike
  ce_opgrp_act      SMALLINT,                               -- 0 = don't activate output group, 1 = activate output group
  ce_reset_pan_alrm SMALLINT,                               -- 0 = don't restart panel alarm relay, 1 = restart panel alarm relay
  ce_ip_enable      SMALLINT,                               -- 0 = don't allow input group suppression, 1 = allow input group
                                                              suppression
  ce_op_enable      SMALLINT,                               -- 0 = don't allow output group activation, 1 = allow output group
                                                              activation
  ce_term_1         SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_2         SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_3         SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_4         SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_5         SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_6         SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_7         SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_8         SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_9         SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_10        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_11        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_12        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_13        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_14        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_15        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_16        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_17        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_18        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_19        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_20        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_21        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_22        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_23        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_24        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_25        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_26        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_27        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_28        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_29        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_30        SMALLINT,                               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
```

```
  ce_term_31            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_32            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_33            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_34            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_35            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_36            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_37            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_38            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_39            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_40            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_41            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_42            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_43            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_44            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_45            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_46            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_47            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_48            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_49            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_50            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_51            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_52            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_53            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_54            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_55            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_56            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_57            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_58            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_59            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_60            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_61            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_62            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_63            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  ce_term_64            SMALLINT,                                   -- 0 = invalid for this terminal index, 1 = valid for this terminal index
  site                  CHAR(32) NOT NULL,                          -- Site Name
  ce_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  ce_checksum           INTEGER NOT NULL,                           -- Record checksum
);
  CREATE UNIQUE INDEX     ce_index ON cardevent (ce_panel_id,  ce_event_number);
  CREATE UNIQUE INDEX     ce_site_name ON cardevent (site, ce_event_name);
```

CREATE TABLE cardevent_save

```
(
  ce_id                 INTEGER NOT NULL,       -- unique database ID
  ce_panel_id           INTEGER,                -- ID of panel this card event pertains to
  ce_event_name         CHAR(32) NOT NULL,      -- name of this event
  ce_event_number       SMALLINT,               -- unique event number for the selected panel
  ce_trigger_type       SMALLINT,               -- 0 = "Card Only", 1 = "Card/PIN Code", 2 = "Card/Keypad Code",
                                                   3 = "Card/PIN/Keypad Code", 4 = "Any Void Card"
  ce_priv_level         SMALLINT,               -- priviledge level required for this event
  ce_access_code        CHAR(4) NULL,           -- keypad code to trigger event
  ce_event_period       SMALLINT,               -- event duration in minutes
  ce_output_group       INTEGER NULL,           -- ID of output group that can be activated
  ce_input_group        INTEGER NULL,           -- ID of input group that can be suppressed
  ce_ip_suppress        SMALLINT,               -- 0 = don't suppress input group, 1 = suppress input group
  ce_door_strike        SMALLINT,               -- 0 = don't operate door strike, 1 = operate door strike
  ce_opgrp_act          SMALLINT,               -- 0 = don't activate output group, 1 = activate output group
  ce_reset_pan_alrm     SMALLINT,               -- 0 = don't restart panel alarm relay, 1 = restart panel alarm relay
  ce_ip_enable          SMALLINT,               -- 0 = don't allow input group suppression, 1 = allow input group
                                                   suppression
  ce_op_enable          SMALLINT,               -- 0 = don't allow output group activation, 1 = allow output group
                                                   activation
  ce_term_1             SMALLINT,               -- 0 = invalid for this terminal index, 1 = valid for this terminal index
```

```
ce_term_2       SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_3       SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_4       SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_5       SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_6       SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_7       SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_8       SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_9       SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_10      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_11      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_12      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_13      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_14      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_15      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_16      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_17      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_18      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_19      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_20      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_21      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_22      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_23      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_24      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_25      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_26      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_27      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_28      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_29      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_30      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_31      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_32      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_33      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_34      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_35      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_36      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_37      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_38      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_39      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_40      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_41      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_42      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_43      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_44      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_45      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_46      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_47      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_48      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_49      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_50      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_51      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_52      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_53      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_54      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_55      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_56      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_57      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_58      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_59      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_60      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_61      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_62      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_63      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
ce_term_64      SMALLINT,       -- 0 = invalid for this terminal index, 1 = valid for this terminal index
site            CHAR(32) NOT NULL,      -- Site Name
ce_guid         uniqueidentifier NOT NULL,      -- Global unique record identifier
```

```
  ce_checksum           INTEGER NOT NULL,                    -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,           -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                   -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                   -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                    -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,           -- Global unique record identifier
  PRIMARY KEY           (ce_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE cardholdersite

```
(
  csi_id                INTEGER IDENTITY NOT FOR REPLICATION
                          PRIMARY KEY,                       -- Unique database ID
  csi_c_id              INTEGER NOT NULL REFERENCES cardholder(c_id)
                          NOT FOR REPLICATION,
  csi_enable            SMALLINT NOT NULL,
  site                  char(32) NOT NULL,
  csi_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  csi_checksum          INTEGER NOT NULL,
  csi_active            SMALLINT                             -- calculated flag based upon start/terminate timestamps
);
  CREATE UNIQUE INDEX     cardholdersite_site ON  cardholdersite(site, csi_c_id);
  CREATE INDEX            cardholdersite_c_id ON  cardholdersite(csi_c_id);
  CREATE INDEX            csi_active ON cardholdersite ( csi_active );
  CREATE INDEX            site_id_enable ON cardholdersite ( site, csi_c_id, csi_enable);
```

CREATE TABLE cardholdersite_save

```
(
  csi_id                INTEGER,                             -- Unique database ID
  csi_c_id              INTEGER NOT NULL,                    -- cardholder
  csi_enable            SMALLINT NOT NULL,
  site                  char(32) NOT NULL,                   -- site name
  csi_guid              uniqueidentifier NOT NULL,           -- Global unique record identifier
  csi_checksum          INTEGER NOT NULL,
  upd_guid              uniqueidentifier NOT NULL,           -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                   -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                   -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                    -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,           -- Global unique record identifier
  csi_active            SMALLINT,                            -- calculated flag based upon start/terminate timestamps
  PRIMARY KEY           NONCLUSTERED ( csi_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE cardholder_save

```
(
  c_partition           INTEGER,                             -- partition ID to which this cardholder belongs
  c_public              SMALLINT,                            -- 1 = cardholder is public, 0 = cardholder is private
  c_id                  INTEGER NOT NULL,                    -- unique database ID
  c_lname               CHAR(25) NOT NULL,                   -- last name
  c_mname               CHAR(25) NULL,                       -- middle name
  c_fname               CHAR(25) NOT NULL,                   -- first name
  c_nick_name           CHAR(25) NULL,                       -- ID, must be unique if not 0
  c_addr                CHAR(64) NULL,                       -- street address
  c_addr1               CHAR(32) NULL,                       -- city
  c_addr2               CHAR(32) NULL,                       -- state
  c_addr3               CHAR(32) NULL,                       -- zip code
  c_phone               CHAR(16) NULL,                       -- phone number
  c_ext                 CHAR(6) NULL,                        -- phone extension
  c_s_timestamp         DATETIME,                            -- start timestamp
```

```
  c_t_timestamp          DATETIME,                                          -- terminate timestamp
  c_card_type            SMALLINT,                                          -- 0 = regular, 1 = visitor
  c_dept_id              INTEGER NULL,                                      -- database ID of department
  c_company_id           INTEGER NULL,                                      -- database ID of company
  c_sponsor_id           INTEGER NULL,                                      -- database ID of sponsor (visitor only)
  c_guard                SMALLINT,                                          -- Can be guard or not
  c_suite                CHAR(32) NULL,                                     -- suit information
  site                   CHAR(32) NOT NULL,                                 -- Site Name
  c_guid                 uniqueidentifier NOT NULL,                         -- Global unique record identifier
  c_checksum             INTEGER NOT NULL,                                  -- Record checksum
  upd_guid               uniqueidentifier NOT NULL,                         -- Guid for update operation
  upd_mode               SMALLINT NOT NULL,                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp          DATETIME NOT NULL,                                 -- Date / Time of the update
  upd_checksum           INTEGER NOT NULL,                                  -- Checksum
  upd_rowguid            uniqueidentifier ROWGUIDCOL DEFAULT
                           newid() UNIQUE NOT NULL,                         -- Global unique record identifier
  c_email                CHAR(64) NULL,                                     -- email address
  c_operator             CHAR(32),                                          -- Operator that last modified record
  c_edit_time            DATETIME,                                          -- Time this record was last edited
  PRIMARY KEY            ( c_partition, c_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE [dbo].[CCTVAlarm]

```
(
  [CAL_ID]               [int] IDENTITY NOT FOR REPLICATION NOT NULL,       -- Unique ID of alarm record
  [CAL_ParentID]         [int] NOT NULL,                                    -- ID of parent record
  [CAL_Name]             [char] (64) NOT NULL,                              -- OPC name of alarm
  [CAL_ServerID]         [int] NOT NULL,                                    -- ID of server record
  [CAL_Exists]           [int] NOT NULL CONSTRAINT [DF_CCTVAlarm_Exists] DEFAULT (1),  -- 0 = No, 1 = Yes, 2 = Use default
  [CAL_Description]      [char] (64) NOT NULL,                              -- Description
  [CAL_Partition]        [int] NOT NULL,                                    -- Partition ID to which this alarm belongs
  [CAL_Public]           [int] NOT NULL,                                    -- 1 = alarm is public, 0 = alarm is private
  site                   CHAR(32) NOT NULL,                                 -- Site Name
  CAL_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  CAL_checksum           INTEGER NOT NULL,                                  -- Record checksum
  CONSTRAINT [PK_CCTVAlarm] PRIMARY KEY CLUSTERED
(
  [CAL_ID]
  ) ON [PRIMARY],
  CONSTRAINT [FK_Alarm_CCTVSwitch] FOREIGN KEY
(
  [CAL_ParentID]
  ) REFERENCES [dbo].[CCTVSwitch] (
  [CSW_ID]
  )  NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVAlarm_CCTVServer] FOREIGN KEY
(
  [CAL_ServerID]
  ) REFERENCES [dbo].[CCTVServer] (
  [CSV_ID]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVAlarm_partition] FOREIGN KEY
(
  [CAL_Partition]
  ) REFERENCES [dbo].[partition] (
  [part_number]
  )  NOT FOR REPLICATION
  ) ON [PRIMARY]
;
```

CREATE TABLE [CCTVAlarm_save]

```
(
  [CAL_ID]              [int] NOT NULL,                          -- Unique ID of alarm record
  [CAL_ParentID]        [int] NOT NULL,                          -- ID of parent record
  [CAL_Name]            [char] (64) NOT NULL,                    -- OPC name of alarm
  [CAL_ServerID]        [int] NOT NULL,                          -- ID of server record
  [CAL_Exists]          [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
  [CAL_Description]     [char] (64) NOT NULL,                    -- Description
  [CAL_Partition]       [int] NOT NULL,                          -- Partition ID to which this alarm belongs
  [CAL_Public]          [int] NOT NULL,                          -- 1 = alarm is public, 0 = alarm is private
  site                  CHAR(32) NOT NULL,                       -- Site Name
  CAL_guid              uniqueidentifier NOT NULL,               -- Global unique record identifier
  CAL_checksum          INTEGER NOT NULL,                        -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,               -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                       -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                       -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                        -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,               -- Global unique record identifier
  CONSTRAINT [PK_CCTVAlarm_save] PRIMARY KEY  CLUSTERED
(
  [CAL_ID], upd_mode desc, upd_guid
)  ON [PRIMARY],
) ON [PRIMARY]
;
```

CREATE TABLE [dbo].[CCTVAuxiliaryCamera]

```
(
  [CAC_ID]              [int] IDENTITY NOT FOR REPLICATION NOT NULL,  -- Unique ID of camera auxiliary record
  [CAC_ParentID]        [int] NOT NULL,                               -- ID of parent record
  [CAC_Name]            [char] (64) NOT NULL,                         -- OPC name of camera auxiliary
  [CAC_ServerID]        [int] NOT NULL,                               -- ID of server record
  [CAC_Exists]          [int] NOT NULL CONSTRAINT
                          [DF_CCTVAuxiliaryCamera_Exists] DEFAULT (1), -- 0 = No, 1 = Yes, 2 = Use default
  [CAC_Description]     [char] (64) NOT NULL,                         -- Description
  [CAC_Partition]       [int] NOT NULL,                               -- Partition ID to which this camera auxiliary belongs
  [CAC_Public]          [int] NOT NULL,                               -- 1 = camera auxiliary is public, 0 = camera auxiliary is
                                                                        private
  site                  CHAR(32) NOT NULL,                            -- Site Name
  CAC_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  CAC_checksum          INTEGER NOT NULL,                             -- Record checksum
  CONSTRAINT [PK_CCTVAuxiliaryCamera] PRIMARY KEY CLUSTERED
(
  [CAC_ID]
  ) ON [PRIMARY],
  CONSTRAINT [FK_CCTVAuxiliaryCamera_CCTVCamera] FOREIGN KEY
(
  [CAC_ParentID]
  ) REFERENCES [dbo].[CCTVCamera] (
  [CCA_ID]
  )  NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVAuxiliaryCamera_CCTVServer] FOREIGN KEY
(
  [CAC_ServerID]
  ) REFERENCES [dbo].[CCTVServer] (
  [CSV_ID]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVAuxiliaryCamera_partition] FOREIGN KEY
(
  [CAC_Partition]
  ) REFERENCES [dbo].[partition] (
  [part_number]
  ) NOT FOR REPLICATION
```

```
  ) ON [PRIMARY]
;
```

CREATE TABLE [CCTVAuxiliaryCamera_save]

```
(
  [CAC_ID]              [int] NOT NULL,                          -- Unique ID of camera auxiliary record
  [CAC_ParentID]        [int] NOT NULL,                          -- ID of parent record
  [CAC_Name]            [char] (64) NOT NULL,                    -- OPC name of camera auxiliary
  [CAC_ServerID]        [int] NOT NULL,                          -- ID of server record
  [CAC_Exists]          [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
  [CAC_Description]     [char] (64) NOT NULL,                    -- Description
  [CAC_Partition]       [int] NOT NULL,                          -- Partition ID to which this camera auxiliary belongs
  [CAC_Public]          [int] NOT NULL,                          -- 1 = camera auxiliary is public, 0 = camera auxiliary is private
  site                  CHAR(32) NOT NULL,                       -- Site Name
  CAC_guid              uniqueidentifier NOT NULL,               -- Global unique record identifier
  CAC_checksum          INTEGER NOT NULL,                        -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,               -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                       -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                       -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                        -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,               -- Global unique record identifier
  CONSTRAINT [PK_CCTVAuxiliaryCamera_save] PRIMARY KEY CLUSTERED
(
  [CAC_ID], upd_mode desc, upd_guid
)  ON [PRIMARY],
) ON [PRIMARY]
;
```

CREATE TABLE [dbo].[CCTVAuxiliarySystem]

```
(
  [CAS_ID]              [int] IDENTITY NOT FOR REPLICATION NOT NULL,   -- Unique ID of switch auxiliary record
  [CAS_ParentID]        [int] NOT NULL,                                -- ID of parent record
  [CAS_Name]            [char] (64) NOT NULL,                          -- OPC name of switch auxiliary
  [CAS_ServerID]        [int] NOT NULL,                                -- ID of server record
  [CAS_Exists]          [int] NOT NULL CONSTRAINT
                          [DF_CCTVAuxiliarySystem_Exists] DEFAULT (1),  -- 0 = No, 1 = Yes, 2 = Use default
  [CAS_Description]     [char] (64) NOT NULL,                          -- Description
  [CAS_Partition]       [int] NOT NULL,                                -- Partition ID to which this switch auxiliary belongs
  [CAS_Public]          [int] NOT NULL,                                -- 1 = switch auxiliary is public, 0 = switch auxiliary is private
  site                  CHAR(32) NOT NULL,                             -- Site Name
  CAS_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
  CAS_checksum          INTEGER NOT NULL,                              -- Record checksum
  CONSTRAINT [PK_CCTVAuxiliarySystem] PRIMARY KEY CLUSTERED
(
  [CAS_ID]
  ) ON [PRIMARY],
  CONSTRAINT [FK_Auxiliaries_CCTVSwitch] FOREIGN KEY
(
  [CAS_ParentID]
  ) REFERENCES [dbo].[CCTVSwitch] (
  [CSW_ID]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVAuxiliarySystem_CCTVServer] FOREIGN KEY
(
  [CAS_ServerID]
  ) REFERENCES [dbo].[CCTVServer] (
  [CSV_ID]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVAuxiliarySystem_partition] FOREIGN KEY
(
  [CAS_Partition]
```

```
  ) REFERENCES [dbo].[partition] (
  [part_number]
  )  NOT FOR REPLICATION
  ) ON [PRIMARY]
;
```

CREATE TABLE [CCTVAuxiliarySystem_save]

```
(
  [CAS_ID]              [int]  NOT NULL,                          -- Unique ID of switch auxiliary record
  [CAS_ParentID]        [int] NOT NULL,                           -- ID of parent record
  [CAS_Name]            [char] (64) NOT NULL,                     -- OPC name of switch auxiliary
  [CAS_ServerID]        [int] NOT NULL,                           -- ID of server record
  [CAS_Exists]          [int] NOT NULL,                           -- 0 = No, 1 = Yes, 2 = Use default
  [CAS_Description]     [char] (64) NOT NULL,                     -- Description
  [CAS_Partition]       [int] NOT NULL,                           -- Partition ID to which this switch auxiliary belongs
  [CAS_Public]          [int] NOT NULL,                           -- 1 = switch auxiliary is public, 0 = switch auxiliary is private
  site                  CHAR(32) NOT NULL,                        -- Site Name
  CAS_guid              uniqueidentifier NOT NULL,                -- Global unique record identifier
  CAS_checksum          INTEGER NOT NULL,                         -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                        -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                        -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                         -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,                -- Global unique record identifier
  CONSTRAINT [PK_CCTVAuxiliarySystem_save] PRIMARY KEY CLUSTERED
(
  [CAS_ID], upd_mode desc, upd_guid
)  ON [PRIMARY],
) ON [PRIMARY]
;
```

CREATE TABLE [CCTVCamera]

```
(
  [CCA_ID]              [int] IDENTITY NOT FOR REPLICATION NOT NULL,  -- Unique ID of camera record
  [CCA_ParentID]        [int] NOT NULL,                               -- ID of parent record
  [CCA_Name]            [char] (64) NOT NULL,                         -- OPC name of camera
  [CCA_ServerID]        [int] NOT NULL,                               -- ID of server record
  [CCA_Exists]          [int] NOT NULL CONSTRAINT
                          [DF_CCTVCamera_Exists] DEFAULT (1),         -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_Description]     [char] (64) NOT NULL,                         -- Description
  [CCA_Partition]       [int] NOT NULL CONSTRAINT
                          [DF_Camera_Partition] DEFAULT (0),          -- Partition ID to which this camera belongs
  [CCA_Public]          [int] NOT NULL,                               -- 1 = camera is public, 0 = camera is private
  [CCA_GeneralString]   [char] (64) NULL,                             -- Camera annotation
  [CCA_TiltExists]      [int] NOT NULL CONSTRAINT
                          [DF_Camera_TiltExists] DEFAULT (1),         -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_PanExists]       [int] NOT NULL CONSTRAINT
                          [DF_Camera_PanExists] DEFAULT (1),          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_ZoomExists]      [int] NOT NULL CONSTRAINT
                          [DF_Camera_ZoomExists] DEFAULT (1),         -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_FocusExists]     [int] NOT NULL CONSTRAINT
                          [DF_Camera_FocusExists] DEFAULT (1),        -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_IrisExists]      [int] NOT NULL CONSTRAINT
                          [DF_Camera_IrisExists] DEFAULT (1),         -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_IrisAutomatic]   [int] NOT NULL CONSTRAINT
                          [DF_Camera_IrisAutomatic] DEFAULT (1),      -- 0 = No, 1 = Yes
  [CCA_LensSpeed]       [int] NOT NULL CONSTRAINT
                          [DF_Camera_LensSpeed] DEFAULT (1),          -- Lens speed
  [CCA_LensSpeedMax]    [int] NOT NULL CONSTRAINT
                          [DF_Camera_LensSpeedMax] DEFAULT (1),       -- Maximun lens speed, 1 = speed not variable
  [CCA_WiperExists]     [int] NOT NULL CONSTRAINT
```

```
                                    [DF_Camera_WiperExists] DEFAULT (1),                -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_WasherExists]              [int] NOT NULL CONSTRAINT
                                    [DF_Camera_WasherExists] DEFAULT (1),               -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_LightExists]               [int] NOT NULL CONSTRAINT
                                    [DF_Camera_LightExists] DEFAULT (1),                -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_StatusExists]              [int] NOT NULL CONSTRAINT
                                    [DF_Camera_StatusExists] DEFAULT (1),               -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_PresetExists]              [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PresetExists] DEFAULT (2),               -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_PresetCount]               [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PresetCount] DEFAULT (2),                -- Number configured in database
    [CCA_PresetMax]                 [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PresetMax] DEFAULT (0),                  -- Number created in namespace, -1 = Use default
    [CCA_PresetPlayExists]          [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PresetPlayExists] DEFAULT (2),           -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_PresetRecordExists]        [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PresetRecordExists] DEFAULT (2),         -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_PresetStopExists]          [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PresetClearExists] DEFAULT (2),          -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_PatternExists]             [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PatternExists] DEFAULT (2),              -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_PatternCount]              [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PatternCount] DEFAULT (0),               -- Number configured in database
    [CCA_PatternMax]                [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PatternMax] DEFAULT (99),                -- Number created in namespace, -1 = Use default
    [CCA_PatternPlayExists]         [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PatternPlayExists] DEFAULT (2),          -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_PatternRecordExists]       [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PatternRecordExists] DEFAULT (2),        -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_PatternStopExists]         [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PatternStopExists] DEFAULT (2),          -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_PatternPauseExists]        [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PatternPauseExists] DEFAULT (2),         -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_PatternRestartExists]      [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PatternRestartExists] DEFAULT (2),       -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_PatternForwardExists]      [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PatternForwardExists] DEFAULT (2),       -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_PatternBackwardExists]     [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PatternBackwardExists] DEFAULT (2),      -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_PatternStepForwardExists]  [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PatternStepForwardExists] DEFAULT (2),   -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_PatternStepBackwardExists] [int] NOT NULL CONSTRAINT
                                    [DF_Camera_PatternStepBackwardExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_AuxiliaryExists]           [int] NOT NULL CONSTRAINT
                                    [DF_Camera_AuxiliaryExists] DEFAULT (2),            -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_AuxiliaryCount]            [int] NOT NULL CONSTRAINT
                                    [DF_Camera_AuxiliaryCount] DEFAULT (0),             -- Number configured in database
    [CCA_AuxiliaryMax]              [int] NOT NULL CONSTRAINT
                                    [DF_Camera_AuxiliaryMax] DEFAULT ((-1)),            -- Number created in namespace, -1 = Use default
    [CCA_AuxiliaryPlayExists]       [int] NOT NULL CONSTRAINT
                                    [DF_Camera_AuxiliaryPlayExists] DEFAULT (2),        -- 0 = No, 1 = Yes, 2 = Use default
    [CCA_AuxiliaryStopExists]       [int] NOT NULL CONSTRAINT
                                    [DF_Camera_AuxiliaryStopExists] DEFAULT (2),        -- 0 = No, 1 = Yes, 2 = Use default
    site                            CHAR(32) NOT NULL,                                  -- Site Name
    CCA_guid                        uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
    CCA_checksum                    INTEGER NOT NULL,                                   -- Record checksum
    CONSTRAINT [PK_Camera] PRIMARY KEY CLUSTERED
(
    [CCA_ID]
    ) ON [PRIMARY],
    CONSTRAINT [FK_Camera_CCTVSwitch] FOREIGN KEY
(
    [CCA_ParentID]
    ) REFERENCES [CCTVSwitch] (
    [CSW_ID]
```

```
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVCamera_CCTVServer] FOREIGN KEY
(
  [CCA_ServerID]
  ) REFERENCES [CCTVServer] (
  [CSV_ID]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVCamera_partition] FOREIGN KEY
(
  [CCA_Partition]
  ) REFERENCES [partition] (
  [part_number]
  ) NOT FOR REPLICATION
  ) ON [PRIMARY]
;
```

CREATE TABLE [CCTVCamera_save]

```
(
  [CCA_ID]                          [int] NOT NULL,          -- Unique ID of camera record
  [CCA_ParentID]                    [int] NOT NULL,          -- ID of parent record
  [CCA_Name]                        [char] (64) NOT NULL,    -- OPC name of camera
  [CCA_ServerID]                    [int] NOT NULL,          -- ID of server record
  [CCA_Exists]                      [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_Description]                 [char] (64) NOT NULL,    -- Description
  [CCA_Partition]                   [int] NOT NULL,          -- Partition ID to which this camera belongs
  [CCA_Public]                      [int] NOT NULL,          -- 1 = camera is public, 0 = camera is private
  [CCA_GeneralString]               [char] (64) NULL,        -- Camera annotation
  [CCA_TiltExists]                  [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_PanExists]                   [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_ZoomExists]                  [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_FocusExists]                 [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_IrisExists]                  [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_IrisAutomatic]               [int] NOT NULL,          -- 0 = No, 1 = Yes
  [CCA_LensSpeed]                   [int] NOT NULL,          -- Lens speed
  [CCA_LensSpeedMax]                [int] NOT NULL,          -- Maximun lens speed, 1 = speed not variable
  [CCA_WiperExists]                 [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_WasherExists]                [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_LightExists]                 [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_StatusExists]                [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_PresetExists]                [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_PresetCount]                 [int] NOT NULL,          -- Number configured in database
  [CCA_PresetMax]                   [int] NOT NULL,          -- Number created in namespace, -1 = Use default
  [CCA_PresetPlayExists]            [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_PresetRecordExists]          [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_PresetStopExists]            [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_PatternExists]               [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_PatternCount]                [int] NOT NULL,          -- Number configured in database
  [CCA_PatternMax]                  [int] NOT NULL,          -- Number created in namespace, -1 = Use default
  [CCA_PatternPlayExists]           [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_PatternRecordExists]         [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_PatternStopExists]           [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_PatternPauseExists]          [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_PatternRestartExists]        [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_PatternForwardExists]        [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_PatternBackwardExists]       [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_PatternStepForwardExists]    [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_PatternStepBackwardExists]   [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_AuxiliaryExists]             [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_AuxiliaryCount]              [int] NOT NULL,          -- Number configured in database
  [CCA_AuxiliaryMax]                [int] NOT NULL,          -- Number created in namespace, -1 = Use default
  [CCA_AuxiliaryPlayExists]         [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  [CCA_AuxiliaryStopExists]         [int] NOT NULL,          -- 0 = No, 1 = Yes, 2 = Use default
  site                              CHAR(32) NOT NULL,       -- Site Name
```

```
  CCA_guid                    uniqueidentifier NOT NULL,          -- Global unique record identifier
  CCA_checksum                INTEGER NOT NULL,                   -- Record checksum
  upd_guid                    uniqueidentifier NOT NULL,          -- Guid for update operation
  upd_mode                    SMALLINT NOT NULL,                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp               DATETIME NOT NULL,                  -- Date / Time of the update
  upd_checksum                INTEGER NOT NULL,                   -- Checksum
  upd_rowguid                 uniqueidentifier ROWGUIDCOL DEFAULT
                                newid() UNIQUE NOT NULL,          -- Global unique record identifier
  CONSTRAINT [PK_Camera_save] PRIMARY KEY CLUSTERED
(
  [CCA_ID], upd_mode desc, upd_guid
)  ON [PRIMARY]
) ON [PRIMARY]
;
```

CREATE TABLE [dbo].[CCTVMacros]

```
(
  [CMA_ID]                    [int] IDENTITY NOT FOR REPLICATION NOT NULL,   -- Unique ID of macro record
  [CMA_ParentID]              [int] NOT NULL,                                -- ID of parent record
  [CMA_Name]                  [char] (64) NOT NULL,                          -- OPC name of macro
  [CMA_ServerID]              [int] NOT NULL,                                -- ID of server record
  [CMA_Exists]                [int] NOT NULL CONSTRAINT
                                [DF_CCTVMacros_Exists] DEFAULT (1),          -- 0 = No, 1 = Yes, 2 = Use default
  [CMA_Description]           [char] (64) NOT NULL,                          -- Description
  [CMA_Partition]             [int] NOT NULL,                                -- Partition ID to which this macro belongs
  [CMA_Public]                [int] NOT NULL,                                -- 1 = macro is public, 0 = macro is private
  site                        CHAR(32) NOT NULL,                             -- Site Name
  CMA_guid                    uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
  CMA_checksum                INTEGER NOT NULL,                              -- Record checksum
  CONSTRAINT [PK_CCTVMacros] PRIMARY KEY CLUSTERED
(
  [CMA_ID]
  ) ON [PRIMARY],
  CONSTRAINT [FK_CCTVMacros_CCTVServer] FOREIGN KEY
(
  [CMA_ServerID]
  ) REFERENCES [dbo].[CCTVServer] (
  [CSV_ID]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVMacros_CCTVSwitch] FOREIGN KEY
(
  [CMA_ParentID]
  ) REFERENCES [dbo].[CCTVSwitch] (
  [CSW_ID]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVMacros_partition] FOREIGN KEY
(
  [CMA_Partition]
  ) REFERENCES [dbo].[partition] (
  [part_number]
  )  NOT FOR REPLICATION
  ) ON [PRIMARY]
;
```

CREATE TABLE [CCTVMacros_save]

```
(
  [CMA_ID]                    [int]  NOT NULL,                   -- Unique ID of macro record
  [CMA_ParentID]              [int] NOT NULL,                    -- ID of parent record
  [CMA_Name]                  [char] (64) NOT NULL,              -- OPC name of macro
  [CMA_ServerID]              [int] NOT NULL,                    -- ID of server record
  [CMA_Exists]                [int] NOT NULL,                    -- 0 = No, 1 = Yes, 2 = Use default
  [CMA_Description]           [char] (64) NOT NULL,              -- Description
  [CMA_Partition]             [int] NOT NULL,                    -- Partition ID to which this macro belongs
```

```
  [CMA_Public]              [int] NOT NULL,                          -- 1 = macro is public, 0 = macro is private
  site                      CHAR(32) NOT NULL,                       -- Site Name
  CMA_guid                  uniqueidentifier NOT NULL,               -- Global unique record identifier
  CMA_checksum              INTEGER NOT NULL,                        -- Record checksum
  upd_guid                  uniqueidentifier NOT NULL,               -- Guid for update operation
  upd_mode                  SMALLINT NOT NULL,                       -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp             DATETIME NOT NULL,                       -- Date / Time of the update
  upd_checksum              INTEGER NOT NULL,                        -- Checksum
  upd_rowguid               uniqueidentifier ROWGUIDCOL DEFAULT
                              newid() UNIQUE NOT NULL,               -- Global unique record identifier
  CONSTRAINT [PK_CCTVMacros_save] PRIMARY KEY CLUSTERED
(
  [CMA_ID], upd_mode desc, upd_guid
)  ON [PRIMARY],
) ON [PRIMARY]
;
```

CREATE TABLE [CCTVMonitor]

```
(
  [CMO_ID]                  [int] IDENTITY NOT FOR REPLICATION
                              NOT NULL,                              -- Unique ID of monitor record
  [CMO_ParentID]            [int] NOT NULL,                          -- ID of parent record
  [CMO_Name]                [char] (64) NOT NULL,                    -- OPC name of monitor
  [CMO_ServerID]            [int] NOT NULL,                          -- ID of server record
  [CMO_Exists]              [int] NOT NULL CONSTRAINT
                              [DF_Monitor_Exists] DEFAULT (1),       -- 0 = No, 1 = Yes, 2 = Use default
  [CMO_Description]         [char] (64) NOT NULL,                    -- Description
  [CMO_Partition]           [int] NOT NULL,                          -- Partition ID to which this monitor belongs
  [CMO_Public]              [int] NOT NULL CONSTRAINT
                              [DF_Monitor_Public] DEFAULT (1),       -- 1 = monitor is public, 0 = monitor is private
  [CMO_GeneralString]       [char] (64) NULL,                        -- monitor annotation
  [CMO_SequenceExists]      [int] NOT NULL CONSTRAINT
                              [DF_Monitor_SequenceExists] DEFAULT (2),     -- 0 = No, 1 = Yes, 2 = Use default
  [CMO_SequenceCount]       [int] NOT NULL CONSTRAINT
                              [DF_Monitor_SequenceCount] DEFAULT (0),      -- Number configured in database
  [CMO_SequenceMax]         [int] NOT NULL CONSTRAINT
                              [DF_Monitor_SequenceMax] DEFAULT ((-1)),     -- Number created in namespace,
                                                                             -1 = Use default
  [CMO_SequencePlayExists]  [int] NOT NULL CONSTRAINT
                              [DF_Monitor_SequencePlayExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [CMO_SequenceRecordExists] [int] NOT NULL CONSTRAINT
                              [DF_Monitor_SequenceRecordExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [CMO_SequenceStopExists]  [int] NOT NULL CONSTRAINT
                              [DF_Monitor_SequenceStopExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [CMO_SequencePauseExists] [int] NOT NULL CONSTRAINT
                              [DF_Monitor_SequencePauseExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [CMO_SequenceRestartExists] [int] NOT NULL CONSTRAINT
                              [DF_Monitor_SequenceRestartExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [CMO_SequenceForwardExists] [int] NOT NULL CONSTRAINT
                              [DF_Monitor_SequenceForwardExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [CMO_SequenceBackwardExists] [int] NOT NULL CONSTRAINT
                              [DF_Monitor_SequenceBackwardExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [CMO_SequenceStepForwardExists] [int] NOT NULL CONSTRAINT
                              [DF_Monitor_SequenceStepForwardExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [CMO_SequenceStepBackwardExists] [int] NOT NULL CONSTRAINT
                              [DF_Monitor_SequenceStepBackwardExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [CMO_SequenceCameraForwardExists] [int] NOT NULL CONSTRAINT
                              [DF_Monitor_SequenceCameraForwardExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
  [CMO_SequenceCameraBackwardExists] [int] NOT NULL CONSTRAINT
                              [DF_Monitor_SequenceCameraBackwardExists] DEFAULT (2),-- 0 = No, 1 = Yes, 2 = Use default
  site                      CHAR(32) NOT NULL,                       -- Site Name
  CMO_guid                  uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  CMO_checksum              INTEGER NOT NULL,                        -- Record checksum
  CONSTRAINT [PK_Monitor] PRIMARY KEY CLUSTERED
```

```
(
   [CMO_ID]
   ) ON [PRIMARY],
   CONSTRAINT [FK_CCTVMonitor_CCTVServer] FOREIGN KEY
(
   [CMO_ServerID]
   ) REFERENCES [CCTVServer] (
   [CSV_ID]
   )  NOT FOR REPLICATION,
   CONSTRAINT [FK_CCTVMonitor_partition] FOREIGN KEY
(
   [CMO_Partition]
   ) REFERENCES [partition] (
   [part_number]
   ) NOT FOR REPLICATION,
   CONSTRAINT [FK_Monitor_CCTVSwitch] FOREIGN KEY
(
   [CMO_ParentID]
   ) REFERENCES [CCTVSwitch] (
   [CSW_ID]
   )  NOT FOR REPLICATION
   ) ON [PRIMARY]
;
```

CREATE TABLE [CCTVMonitor_save]

```
(
   [CMO_ID]                              [int] NOT NULL,                          -- Unique ID of monitor record
   [CMO_ParentID]                        [int] NOT NULL,                          -- ID of parent record
   [CMO_Name]                            [char] (64) NOT NULL,                    -- OPC name of monitor
   [CMO_ServerID]                        [int] NOT NULL,                          -- ID of server record
   [CMO_Exists]                          [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
   [CMO_Description]                     [char] (64) NOT NULL,                    -- Description
   [CMO_Partition]                       [int] NOT NULL,                          -- Partition ID to which this monitor belongs
   [CMO_Public]                          [int] NOT NULL,                          -- 1 = monitor is public, 0 = monitor is private
   [CMO_GeneralString]                   [char] (64) NULL,                        -- monitor annotation
   [CMO_SequenceExists]                  [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
   [CMO_SequenceCount]                   [int] NOT NULL,                          -- Number configured in database
   [CMO_SequenceMax]                     [int] NOT NULL,                          -- Number created in namespace, -1 = Use default
   [CMO_SequencePlayExists]              [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
   [CMO_SequenceRecordExists]            [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
   [CMO_SequenceStopExists]              [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
   [CMO_SequencePauseExists]             [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
   [CMO_SequenceRestartExists]           [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
   [CMO_SequenceForwardExists]           [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
   [CMO_SequenceBackwardExists]          [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
   [CMO_SequenceStepForwardExists]       [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
   [CMO_SequenceStepBackwardExists]      [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
   [CMO_SequenceCameraForwardExists]     [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
   [CMO_SequenceCameraBackwardExists]    [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
   site                                  CHAR(32) NOT NULL,                       -- Site Name
   CMO_guid                              uniqueidentifier NOT NULL,               -- Global unique record identifier
   CMO_checksum                          INTEGER NOT NULL,                        -- Record checksum
   upd_guid                              uniqueidentifier NOT NULL,               -- Guid for update operation
   upd_mode                              SMALLINT NOT NULL,                       -- Update mode (0 - delete, 1 - edit, 2 - new)
   upd_timestamp                         DATETIME NOT NULL,                       -- Date / Time of the update
   upd_checksum                          INTEGER NOT NULL,                        -- Checksum
   upd_rowguid                           uniqueidentifier ROWGUIDCOL DEFAULT
                                            newid() UNIQUE NOT NULL,              -- Global unique record identifier
   CONSTRAINT [PK_Monitor_save] PRIMARY KEY CLUSTERED
(
   [CMO_ID], upd_mode desc, upd_guid
)  ON [PRIMARY],
) ON [PRIMARY]
;
```

CREATE TABLE [dbo].[CCTVPattern]

```
(
  [CPA_ID]                [int] IDENTITY NOT FOR REPLICATION NOT NULL,   -- Unique ID of pattern record
  [CPA_ParentID]          [int] NOT NULL,                                -- ID of parent record
  [CPA_Name]              [char] (64) NOT NULL,                          -- OPC name of pattern
  [CPA_ServerID]          [int] NOT NULL,                                -- ID of server record
  [CPA_Exists]            [int] NOT NULL CONSTRAINT
                            [DF_CCTVPattern_Exists] DEFAULT (1),         -- 0 = No, 1 = Yes, 2 = Use default
  [CPA_Description]       [char] (64) NOT NULL,                          -- Description
  [CPA_Partition]         [int] NOT NULL,                                -- Partition ID to which this pattern belongs
  [CPA_Public]            [int] NOT NULL,                                -- 1 = pattern is public, 0 = pattern is private
  site                    CHAR(32) NOT NULL,                             -- Site Name
  CPA_guid                uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
  CPA_checksum            INTEGER NOT NULL,                              -- Record checksum
  CONSTRAINT [PK_CCTVPattern] PRIMARY KEY CLUSTERED
(
  [CPA_ID]
  ) ON [PRIMARY],
  CONSTRAINT [FK_CCTVPattern_CCTVServer] FOREIGN KEY
(
  [CPA_ServerID]
  ) REFERENCES [dbo].[CCTVServer] (
  [CSV_ID]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVPattern_partition] FOREIGN KEY
(
  [CPA_Partition]
  ) REFERENCES [dbo].[partition] (
  [part_number]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_Pattern_Camera] FOREIGN KEY
(
  [CPA_ParentID]
  ) REFERENCES [dbo].[CCTVCamera] (
  [CCA_ID]
  )  NOT FOR REPLICATION
  ) ON [PRIMARY]
;
```

CREATE TABLE [CCTVPattern_save]

```
(
  [CPA_ID]                [int] NOT NULL,                                -- Unique ID of pattern record
  [CPA_ParentID]          [int] NOT NULL,                                -- ID of parent record
  [CPA_Name]              [char] (64) NOT NULL,                          -- OPC name of pattern
  [CPA_ServerID]          [int] NOT NULL,                                -- ID of server record
  [CPA_Exists]            [int] NOT NULL,                                -- 0 = No, 1 = Yes, 2 = Use default
  [CPA_Description]       [char] (64) NOT NULL,                          -- Description
  [CPA_Partition]         [int] NOT NULL,                                -- Partition ID to which this pattern belongs
  [CPA_Public]            [int] NOT NULL,                                -- 1 = pattern is public, 0 = pattern is private
  site                    CHAR(32) NOT NULL,                             -- Site Name
  CPA_guid                uniqueidentifier NOT NULL,                     -- Global unique record identifier
  CPA_checksum            INTEGER NOT NULL,                              -- Record checksum
  upd_guid                uniqueidentifier NOT NULL,                     -- Guid for update operation
  upd_mode                SMALLINT NOT NULL,                             -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp           DATETIME NOT NULL,                             -- Date / Time of the update
  upd_checksum            INTEGER NOT NULL,                              -- Checksum
  upd_rowguid             uniqueidentifier ROWGUIDCOL DEFAULT
                            newid() UNIQUE NOT NULL,                     -- Global unique record identifier
  CONSTRAINT [PK_CCTVPattern_save] PRIMARY KEY CLUSTERED
(
  [CPA_ID], upd_mode desc, upd_guid
)  ON [PRIMARY],
) ON [PRIMARY]
;
```

CREATE TABLE [dbo].[CCTVPreset]

```
(
  [CPR_ID]              [int] IDENTITY NOT FOR REPLICATION NOT NULL,   -- Unique ID of preset record
  [CPR_ParentID]        [int] NOT NULL,                                -- ID of parent record
  [CPR_Name]            [char] (64) NOT NULL,                          -- OPC name of preset
  [CPR_ServerID]        [int] NOT NULL,                                -- ID of server record
  [CPR_Exists]          [int] NOT NULL CONSTRAINT
                          [DF_CCTVPreset_Exists] DEFAULT (1),          -- 0 = No, 1 = Yes, 2 = Use default
  [CPR_Description]     [char] (64) NOT NULL,                          -- Description
  [CPR_Partition]       [int] NOT NULL,                                -- Partition ID to which this preset belongs
  [CPR_Public]          [int] NOT NULL,                                -- 1 = preset is public, 0 = preset is private
  site                  CHAR(32) NOT NULL,                             -- Site Name
  CPR_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
  CPR_checksum          INTEGER NOT NULL,                              -- Record checksum
  CONSTRAINT [PK_CCTVPreset] PRIMARY KEY CLUSTERED
(
  [CPR_ID]
  ) ON [PRIMARY],
  CONSTRAINT [FK_CCTVPreset_CCTVCamera] FOREIGN KEY
(
  [CPR_ParentID]
  ) REFERENCES [dbo].[CCTVCamera] (
  [CCA_ID]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVPreset_CCTVServer] FOREIGN KEY
(
  [CPR_ServerID]
  ) REFERENCES [dbo].[CCTVServer] (
  [CSV_ID]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVPreset_partition] FOREIGN KEY
(
  [CPR_Partition]
  ) REFERENCES [dbo].[partition] (
  [part_number]
  )NOT FOR REPLICATION
  ) ON [PRIMARY]
;
```

CREATE TABLE [CCTVPreset_save]

```
(
  [CPR_ID]              [int]  NOT NULL,                               -- Unique ID of preset record
  [CPR_ParentID]        [int] NOT NULL,                                -- ID of parent record
  [CPR_Name]            [char] (64) NOT NULL,                          -- OPC name of preset
  [CPR_ServerID]        [int] NOT NULL,                                -- ID of server record
  [CPR_Exists]          [int] NOT NULL,                                -- 0 = No, 1 = Yes, 2 = Use default
  [CPR_Description]     [char] (64) NOT NULL,                          -- Description
  [CPR_Partition]       [int] NOT NULL,                                -- Partition ID to which this preset belongs
  [CPR_Public]          [int] NOT NULL,                                -- 1 = preset is public, 0 = preset is private
  site                  CHAR(32) NOT NULL,                             -- Site Name
  CPR_guid              uniqueidentifier NOT NULL,                     -- Global unique record identifier
  CPR_checksum          INTEGER NOT NULL,                              -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                     -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                             -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                             -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                              -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,                     -- Global unique record identifier
  CONSTRAINT [PK_CCTVPreset_save] PRIMARY KEY CLUSTERED
(
  [CPR_ID], upd_mode desc, upd_guid
)  ON [PRIMARY],
) ON [PRIMARY]
;
```

CREATE TABLE [dbo].[CCTVSequence]

```
(
  [CSE_ID]          [int] IDENTITY NOT FOR REPLICATION NOT NULL,   -- Unique ID of sequence record
  [CSE_ParentID]    [int] NOT NULL,                                -- ID of parent record
  [CSE_Name]        [char] (64) NOT NULL,                          -- OPC name of sequence
  [CSE_ServerID]    [int] NOT NULL,                                -- ID of server record
  [CSE_Exists]      [int] NOT NULL CONSTRAINT
                      [DF_CCTVSequence_Exists] DEFAULT (1),        -- 0 = No, 1 = Yes, 2 = Use default
  [CSE_Description] [char] (64) NOT NULL,                          -- Description
  [CSE_Partition]   [int] NOT NULL,                                -- Partition ID to which this sequence belongs
  [CSE_Public]      [int] NOT NULL,                                -- 1 = sequence is public, 0 = sequence is private
  site              CHAR(32) NOT NULL,                             -- Site Name
  CSE_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
  CSE_checksum      INTEGER NOT NULL,                              -- Record checksum
  CONSTRAINT [PK_CCTVSequence] PRIMARY KEY CLUSTERED
(
  [CSE_ID]
  ) ON [PRIMARY],
  CONSTRAINT [FK_CCTVSequence_CCTVMonitor] FOREIGN KEY
(
  [CSE_ParentID]
  ) REFERENCES [dbo].[CCTVMonitor] (
  [CMO_ID]
  )  NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVSequence_CCTVServer] FOREIGN KEY
(
  [CSE_ServerID]
  ) REFERENCES [dbo].[CCTVServer] (
  [CSV_ID]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVSequence_partition] FOREIGN KEY
(
  [CSE_Partition]
  ) REFERENCES [dbo].[partition] (
  [part_number]
  )NOT FOR REPLICATION
  ) ON [PRIMARY]
;
```

CREATE TABLE [CCTVSequence_save]

```
(
  [CSE_ID]          [int] NOT NULL,                                -- Unique ID of sequence record
  [CSE_ParentID]    [int] NOT NULL,                                -- ID of parent record
  [CSE_Name]        [char] (64) NOT NULL,                          -- OPC name of sequence
  [CSE_ServerID]    [int] NOT NULL,                                -- ID of server record
  [CSE_Exists]      [int] NOT NULL,                                -- 0 = No, 1 = Yes, 2 = Use default
  [CSE_Description] [char] (64) NOT NULL,                          -- Description
  [CSE_Partition]   [int] NOT NULL,                                -- Partition ID to which this sequence belongs
  [CSE_Public]      [int] NOT NULL,                                -- 1 = sequence is public, 0 = sequence is private
  site              CHAR(32) NOT NULL,                             -- Site Name
  CSE_guid          uniqueidentifier NOT NULL,                     -- Global unique record identifier
  CSE_checksum      INTEGER NOT NULL,                              -- Record checksum
  upd_guid          uniqueidentifier NOT NULL,                     -- Guid for update operation
  upd_mode          SMALLINT NOT NULL,                             -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp     DATETIME NOT NULL,                             -- Date / Time of the update
  upd_checksum      INTEGER NOT NULL,                              -- Checksum
  upd_rowguid       uniqueidentifier ROWGUIDCOL DEFAULT
                      newid() UNIQUE NOT NULL,                     -- Global unique record identifier
  CONSTRAINT [PK_CCTVSequence_save] PRIMARY KEY CLUSTERED
(
  [CSE_ID], upd_mode desc, upd_guid
)  ON [PRIMARY],
) ON [PRIMARY]
;
```

CREATE TABLE [CCTVServer]

```
(
  [CSV_ID]              [int] IDENTITY NOT FOR REPLICATION NOT NULL,     -- Unique ID of server record
  [CSV_ParentID]        [int] NOT NULL,                                  -- ID of parent record
  [CSV_Name]            [char] (64) NOT NULL,                            -- OPC Name - defaults to description
  [CSV_ServerID]        [int] NOT NULL,                                  -- Same as CSV_ID
  [CSV_Exists]          [int] NOT NULL CONSTRAINT [DF_CCTVServer_Exists]
                          DEFAULT (1),                                   -- 0 = No, 1 = Yes, 2 = Use default
  [CSV_Description]     [char] (64) NOT NULL,                            -- Description - must be unique
  [CSV_Partition]       [int] NOT NULL,                                  -- Partition ID to which this server belongs
  [CSV_Public]          [int] NOT NULL,                                  -- 1 = server is public, 0 = server is private
  [CSV_PCName]          [char] (64) NOT NULL,                            -- PC Name of server computer
  [CSV_ProgID]          [char] (64) NOT NULL,                            -- Registry name of CCTV Server
  site                  CHAR(32) NOT NULL,                               -- Site Name
  CSV_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,     -- Global unique record identifier
  CSV_checksum          INTEGER NOT NULL,                                -- Record checksum
  CONSTRAINT [PK_CCTVServer] PRIMARY KEY CLUSTERED
(
  [CSV_ID]
  ) ON [PRIMARY],
  CONSTRAINT [FK_CCTVServer_partition] FOREIGN KEY
(
  [CSV_Partition]
  ) REFERENCES [partition] (
  [part_number]
  ) NOT FOR REPLICATION,
  ) ON [PRIMARY]
  ;
```

CREATE TABLE [CCTVServer_save]

```
(
  [CSV_ID]              [int]  NOT NULL,                                 -- Unique ID of server record
  [CSV_ParentID]        [int] NOT NULL,                                  -- ID of parent record
  [CSV_Name]            [char] (64) NOT NULL,                            -- OPC Name - defaults to description
  [CSV_ServerID]        [int] NOT NULL,                                  -- Same as CSV_ID
  [CSV_Exists]          [int] NOT NULL,                                  -- 0 = No, 1 = Yes, 2 = Use default
  [CSV_Description]     [char] (64) NOT NULL,                            -- Description - must be unique
  [CSV_Partition]       [int] NOT NULL,                                  -- Partition ID to which this server belongs
  [CSV_Public]          [int] NOT NULL,                                  -- 1 = server is public, 0 = server is private
  [CSV_PCName]          [char] (64) NOT NULL,                            -- PC Name of server computer
  [CSV_ProgID]          [char] (64) NOT NULL,                            -- Registry name of CCTV Server
  site                  CHAR(32) NOT NULL,                               -- Site Name
  CSV_guid              uniqueidentifier NOT NULL,                       -- Global unique record identifier
  CSV_checksum          INTEGER NOT NULL,                                -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                       -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                               -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                               -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                               -- Global unique record identifier
  CONSTRAINT [PK_CCTVServer_save] PRIMARY KEY CLUSTERED
(
  [CSV_ID], upd_mode desc, upd_guid
)  ON [PRIMARY],
) ON [PRIMARY]
;
```

CREATE TABLE [CCTVSwitch]

```
(
  [CSW_ID]              [int] IDENTITY NOT FOR REPLICATION NOT NULL,     -- Unique ID of switch record
  [CSW_ParentID]        [int] NOT NULL,                                  -- ID of parent record
  [CSW_Name]            [char] (64) NOT NULL,                            -- OPC name of switch
```

```
[CSW_ServerID]          [int] NOT NULL,                                  -- ID of server record
[CSW_Exists]            [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_Exists] DEFAULT (1),            -- 0 = No, 1 = Yes, 2 = Use default
[CSW_Description]       [char] (64) NOT NULL,                            -- Description
[CSW_Partition]         [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_Partition] DEFAULT (0),         -- Partition ID to which this switch belongs
[CSW_Public]            [int] NOT NULL CONSTRAINT [
                          DF_CCTVSwitch_Public] DEFAULT (1),             -- 1 = switch is public, 0 = switch is private
[CSW_Priority]          [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_Priority] DEFAULT (0),          -- Not used
[CSW_CCTVProtocolType]  [char] (64) NOT NULL,                            -- CCTV manufacturer protocol type
[CSW_Type]              [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_Type] DEFAULT (0),              -- 1 = Serial comms, 2 = TCP/IP
[CSW_Port]              [char] (64) NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_Port] DEFAULT ('COM1'),         -- Serial port switch is connected to
[CSW_Baudrate]          [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_Baudrate] DEFAULT (9600),       -- Baudrate
[CSW_DataBits]          [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_DataBits] DEFAULT (8),          -- Word size
[CSW_Parity]            [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_Parity] DEFAULT (0),            -- Parity setting
[CSW_StopBits]          [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_StopBits] DEFAULT (1),          -- Stop bits
[CSW_IPAddress]         [char] (64) NULL,                                -- IP address or computer name
[CSW_AlarmExists]       [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_AlarmExists] DEFAULT (2),       -- 0 = No, 1 = Yes, 2 = Use default
[CSW_AlarmCount]        [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_AlarmCount] DEFAULT (0),        -- Number configured in database
[CSW_AlarmMax]          [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_AlarmMax] DEFAULT ((-1)),       -- Number created in namespace,
                                                                           -1 = Use default
[CSW_AlarmPlayExists]   [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_AlarmPlayExists] DEFAULT (2),   -- 0 = No, 1 = Yes, 2 = Use default
[CSW_AlarmStopExists]   [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_AlarmStopExists] DEFAULT (2),   -- 0 = No, 1 = Yes, 2 = Use default
[CSW_AuxiliaryExists]   [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_AuxiliaryExists] DEFAULT (2),   -- 0 = No, 1 = Yes, 2 = Use default
[CSW_AuxiliaryCount]    [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_AuxiliaryCount] DEFAULT (0),    -- Number configured in database
[CSW_AuxiliaryMax]      [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_AuxiliaryMax] DEFAULT ((-1)),   -- Number created in namespace,
                                                                           -1 = Use default
[CSW_AuxiliaryPlayExists]  [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_AuxiliaryPlayExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
[CSW_AuxiliaryStopExists]  [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_AuxiliaryStopExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
[CSW_MacroExists]       [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_MacroExists] DEFAULT (2),       -- 0 = No, 1 = Yes, 2 = Use default
[CSW_MacroCount]        [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_MacroCount] DEFAULT (2),        -- Number configured in database
[CSW_MacroMax]          [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_MacroMax] DEFAULT ((-1)),       -- Number created in namespace,
                                                                           -1 = Use default
[CSW_MacroPlayExists]   [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_MacroPlayExists] DEFAULT (2),   -- 0 = No, 1 = Yes, 2 = Use default
[CSW_MacroRecordExists] [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_MacroRecordExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
[CSW_MacroStopExists]   [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_MacroStopExists] DEFAULT (2),   -- 0 = No, 1 = Yes, 2 = Use default
[CSW_MacroPauseExists]  [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_MacroPauseExists] DEFAULT (2),  -- 0 = No, 1 = Yes, 2 = Use default
[CSW_MacroRestartExists] [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_MacroRestartExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourExists]        [int] NOT NULL CONSTRAINT
                          [DF_CCTVSwitch_TourExists] DEFAULT (2),        -- 0 = No, 1 = Yes, 2 = Use default
```

```
[CSW_TourCount]                 [int] NOT NULL CONSTRAINT
                                   [DF_CCTVSwitch_TourCount] DEFAULT (0),                    -- Number configured in database
[CSW_TourMax]                   [int] NOT NULL CONSTRAINT
                                   [DF_CCTVSwitch_TourMax] DEFAULT ((-1)),                   -- Number created in namespace,
                                                                                               -1 = Use default
[CSW_TourPlayExists]               [int] NOT NULL CONSTRAINT [DF_CCTVSwitch_TourPlayExists]
                                      DEFAULT (2),                                           -- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourRecordExists]             [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_TourRecordExists] DEFAULT (2),          -- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourStopExists]               [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_TourStopExists] DEFAULT (2),            -- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourPauseExists]              [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_TourPauseExists] DEFAULT (2),           -- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourRestartExists]            [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_TourRestartExists] DEFAULT (2),         -- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourForwardExists]            [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_TourForwardExists] DEFAULT (2),         -- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourBackwardExists]           [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_TourBackwardExists] DEFAULT (2),        -- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourStepForwardExists]        [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_TourStepForwardExists] DEFAULT (2),     -- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourStepBackwardExists]       [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_StepBackwardExists] DEFAULT (2),        -- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourCameraForwardExists]      [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_TourCameraForwardExists] DEFAULT (2),   -- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourCameraBackwardExists]     [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_TourCameraBackwardExists] DEFAULT (2),  -- 0 = No, 1 = Yes, 2 = Use default
[CSW_MonitorCount]                 [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_MonitorCount] DEFAULT (0),              -- Number configured in database
[CSW_MonitorMax]                   [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_MonitorMax] DEFAULT ((-1)),             -- Number created in namespace,
                                                                                               -1 = Use default
[CSW_SequenceExists]               [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_SequenceExists] DEFAULT (2),            -- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequenceMax]                  [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_SequenceMax] DEFAULT ((-1)),            -- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequencePlayExists]           [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_SequencePlayExists] DEFAULT (2),        -- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequenceRecordExists]         [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_SequenceRecordExists] DEFAULT (2),      -- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequenceStopExists]           [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_SequenceStopExists] DEFAULT (2),        -- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequencePauseExists]          [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_SequencePauseExists] DEFAULT (2),       -- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequenceRestartExists]        [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_SequenceRestartExists] DEFAULT (2),     -- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequenceStepForwardExists]    [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_SequenceStepForwardExists] DEFAULT (2), -- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequenceStepBackwardExists]   [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_SequenceStepBackwardExists] DEFAULT (2),-- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequenceCameraForwardExists]  [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_SequenceCameraForwardExists] DEFAULT (2),-- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequenceCameraBackwardExists] [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_SequenceCameraBackwardExists] DEFAULT (2),-- 0 = No, 1 = Yes, 2 = Use default
[CSW_CameraCount]                  [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_CameraCount] DEFAULT (0),               -- Number configured in database
[CSW_CameraMax]                    [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_CameraMax] DEFAULT ((-1)),              -- Number created in namespace,
                                                                                               -1 = Use default
[CSW_PresetExists]                 [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_PresetExists] DEFAULT (2),              -- 0 = No, 1 = Yes, 2 = Use default
[CSW_PresetMax]                    [int] NOT NULL CONSTRAINT
                                      [DF_CCTVSwitch_PresetMax] DEFAULT ((-1)),              -- Number created in namespace,
                                                                                               -1 = Use default
[CSW_PresetPlayExists]             [int] NOT NULL CONSTRAINT
```

```
    [DF_CCTVSwitch_PresetPlayExists] DEFAULT (2),                -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_PresetRecordExists]      [int] NOT NULL CONSTRAINT
    [DF_CCTVSwitch_PresetRecordExists] DEFAULT (2),              -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_PresetStopExists]        [int] NOT NULL CONSTRAINT
    [DF_CCTVSwitch_PresetStopExists] DEFAULT (2),                -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_PatternExists]           [int] NOT NULL CONSTRAINT
    [DF_CCTVSwitch_PatternExists] DEFAULT (2),                   -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_PatternMax]              [int] NOT NULL CONSTRAINT
    [DF_CCTVSwitch_PatternMax] DEFAULT ((-1)),                   -- Number created in namespace,
                                                                    -1 = Use default
  [CSW_PatternPlayExists]       [int] NOT NULL CONSTRAINT
    [DF_CCTVSwitch_PatternPlayExists] DEFAULT (2),               -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_PatternRecordExists]     [int] NOT NULL CONSTRAINT
    [DF_CCTVSwitch_PatternRecordExists] DEFAULT (2),             -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_PatternStopExists]       [int] NOT NULL CONSTRAINT
    [DF_CCTVSwitch_PatternStopExists] DEFAULT (2),               -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_PatternPauseExists]      [int] NOT NULL CONSTRAINT
    [DF_CCTVSwitch_PatternPauseExists] DEFAULT (2),              -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_PatternRestartExists]    [int] NOT NULL CONSTRAINT
    [DF_CCTVSwitch_PatternRestartExists] DEFAULT (2),            -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_PatternStepForwardExists]  [int] NOT NULL CONSTRAINT
    [DF_CCTVSwitch_PatternStepForwardExists] DEFAULT (2),        -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_PatternStepBackwardExists] [int] NOT NULL CONSTRAINT
    [DF_CCTVSwitch_PatternStepBackwardExists] DEFAULT (2),       -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_CameraAuxiliaryExists]   [int] NOT NULL CONSTRAINT
    [DF_CCTVSwitch_CameraAuxiliaryExists] DEFAULT (2),           -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_CameraAuxiliaryMax]      [int] NOT NULL CONSTRAINT
    [DF_CCTVSwitch_CameraAuxiliaryMax] DEFAULT ((-1)),           -- Number created in namespace,
                                                                    -1 = Use default
  [CSW_CameraAuxiliaryPlayExists]   [int] NOT NULL CONSTRAINT
    [DF_CCTVSwitch_CameraAuxiliaryPlayExists] DEFAULT (2),       -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_CameraAuxiliaryStopExists]   [int] NOT NULL CONSTRAINT
    [DF_CCTVSwitch_CameraAuxiliaryStopExists] DEFAULT (2),       -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_TimeOut]                 [int] not NULL,                  -- TimeOut time for communication with matrix
  site                          CHAR(32) NOT NULL,               -- Site Name
  CSW_guid                      uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  CSW_checksum                  INTEGER NOT NULL,                -- Record checksum
  CONSTRAINT [PK_CCTVSwitch] PRIMARY KEY CLUSTERED
(
  [CSW_ID]
  ) ON [PRIMARY],               --
  CONSTRAINT [FK_CCTVSwitch_CCTVServer] FOREIGN KEY
(
  [CSW_ServerID]
  ) REFERENCES [CCTVServer] (
  [CSV_ID]
  ) NOT FOR REPLICATION,        --
  CONSTRAINT [FK_CCTVSwitch_partition] FOREIGN KEY
(
  [CSW_Partition]
  ) REFERENCES [partition] (
  [part_number]
  ) NOT FOR REPLICATION,        --
  ) ON [PRIMARY]
  ;
```

CREATE TABLE [CCTVSwitch_save]

```
(
  [CSW_ID]                      [int]  NOT NULL,               -- Unique ID of switch record
  [CSW_ParentID]                [int] NOT NULL,                -- ID of parent record
  [CSW_Name]                    [char] (64) NOT NULL,          -- OPC name of switch
  [CSW_ServerID]                [int] NOT NULL,                -- ID of server record
  [CSW_Exists]                  [int] NOT NULL,                -- 0 = No, 1 = Yes, 2 = Use default
```

[CSW_Description]	[char] (64) NOT NULL,	-- Description
[CSW_Partition]	[int] NOT NULL,	-- Partition ID to which this switch belongs
[CSW_Public]	[int] NOT NULL,	-- 1 = switch is public, 0 = switch is private
[CSW_Priority]	[int] NOT NULL,	-- Not used
[CSW_CCTVProtocolType]	[char] (64) NOT NULL,	-- CCTV manufacturer protocol type
[CSW_Type]	[int] NOT NULL,	-- 1 = Serial comms, 2 = TCP/IP
[CSW_Port]	[char] (64) NOT NULL,	-- Serial port switch is connected to
[CSW_Baudrate]	[int] NOT NULL,	-- Baudrate
[CSW_DataBits]	[int] NOT NULL,	-- Word size
[CSW_Parity]	[int] NOT NULL,	-- Parity setting
[CSW_StopBits]	[int] NOT NULL,	-- Stop bits
[CSW_IPAddress]	[char] (64) NULL,	-- IP address or computer name
[CSW_AlarmExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_AlarmCount]	[int] NOT NULL,	-- Number configured in database
[CSW_AlarmMax]	[int] NOT NULL,	-- Number created in namespace, -1 = Use default
[CSW_AlarmPlayExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_AlarmStopExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_AuxiliaryExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_AuxiliaryCount]	[int] NOT NULL,	-- Number configured in database
[CSW_AuxiliaryMax]	[int] NOT NULL,	-- Number created in namespace, -1 = Use default
[CSW_AuxiliaryPlayExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_AuxiliaryStopExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_MacroExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_MacroCount]	[int] NOT NULL,	-- Number configured in database
[CSW_MacroMax]	[int] NOT NULL,	-- Number created in namespace, -1 = Use default
[CSW_MacroPlayExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_MacroRecordExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_MacroStopExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_MacroPauseExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_MacroRestartExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourCount]	[int] NOT NULL,	-- Number configured in database
[CSW_TourMax]	[int] NOT NULL,	-- Number created in namespace, -1 = Use default
[CSW_TourPlayExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourRecordExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourStopExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourPauseExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourRestartExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourForwardExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourBackwardExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourStepForwardExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourStepBackwardExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourCameraForwardExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_TourCameraBackwardExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_MonitorCount]	[int] NOT NULL,	-- Number configured in database
[CSW_MonitorMax]	[int] NOT NULL,	-- Number created in namespace, -1 = Use default
[CSW_SequenceExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequenceMax]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequencePlayExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequenceRecordExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequenceStopExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequencePauseExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequenceRestartExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequenceStepForwardExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequenceStepBackwardExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequenceCameraForwardExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_SequenceCameraBackwardExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_CameraCount]	[int] NOT NULL,	-- Number configured in database
[CSW_CameraMax]	[int] NOT NULL,	-- Number created in namespace, -1 = Use default
[CSW_PresetExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_PresetMax]	[int] NOT NULL,	-- Number created in namespace, -1 = Use default
[CSW_PresetPlayExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_PresetRecordExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_PresetStopExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default
[CSW_PatternExists]	[int] NOT NULL,	-- 0 = No, 1 = Yes, 2 = Use default

```
  [CSW_PatternMax]                [int] NOT NULL,                -- Number created in namespace, -1 = Use default
  [CSW_PatternPlayExists]         [int] NOT NULL,                -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_PatternRecordExists]       [int] NOT NULL,                -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_PatternStopExists]         [int] NOT NULL,                -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_PatternPauseExists]        [int] NOT NULL,                -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_PatternRestartExists]      [int] NOT NULL,                -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_PatternStepForwardExists]  [int] NOT NULL,                -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_PatternStepBackwardExists] [int] NOT NULL,                -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_CameraAuxiliaryExists]     [int] NOT NULL,                -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_CameraAuxiliaryMax]        [int] NOT NULL,                -- Number created in namespace, -1 = Use default
  [CSW_CameraAuxiliaryPlayExists] [int] NOT NULL,                -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_CameraAuxiliaryStopExists] [int] NOT NULL,                -- 0 = No, 1 = Yes, 2 = Use default
  [CSW_TimeOut]                   [int] not NULL,                -- TimeOut time for communication with matrix
  site                            CHAR(32) NOT NULL,             -- Site Name
  CSW_guid                        uniqueidentifier NOT NULL,     -- Global unique record identifier
  CSW_checksum                    INTEGER NOT NULL,              -- Record checksum
  upd_guid                        uniqueidentifier NOT NULL,     -- Guid for update operation
  upd_mode                        SMALLINT NOT NULL,             -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                   DATETIME NOT NULL,             -- Date / Time of the update
  upd_checksum                    INTEGER NOT NULL,              -- Checksum
  upd_rowguid                     uniqueidentifier ROWGUIDCOL DEFAULT
                                      newid() UNIQUE NOT NULL,   -- Global unique record identifier
  CONSTRAINT [PK_CCTVSwitch_save] PRIMARY KEY  CLUSTERED
(
  [CSW_ID], upd_mode desc, upd_guid
)  ON [PRIMARY],   --
) ON [PRIMARY]
;
```

CREATE TABLE [dbo].[CCTVTour]

```
(
  [CTO_ID]          [int] IDENTITY NOT FOR REPLICATION NOT NULL,                -- Unique ID of tour record
  [CTO_ParentID]    [int] NOT NULL,                                             -- ID of parent record
  [CTO_Name]        [char] (64) NOT NULL,                                       -- OPC name of tour
  [CTO_ServerID]    [int] NOT NULL,                                             -- ID of server record
  [CTO_Exists]      [int] NOT NULL CONSTRAINT [DF_CCTVTour_Exists] DEFAULT (1), -- 0 = No, 1 = Yes, 2 = Use default
  [CTO_Description] [char] (64) NOT NULL,                                       -- Description
  [CTO_Partition]   [int] NOT NULL,                                             -- Partition ID to which this tour belongs
  [CTO_Public]      [int] NOT NULL,                                             -- 1 = tour is public, 0 = tour is private
  site              CHAR(32) NOT NULL,                                          -- Site Name
  CTO_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                -- Global unique record identifier
  CTO_checksum      INTEGER NOT NULL,                                           -- Record checksum
  CONSTRAINT [PK_CCTVTour] PRIMARY KEY CLUSTERED
(
  [CTO_ID]
  ) ON [PRIMARY],
  CONSTRAINT [FK_CCTVTour_CCTVServer] FOREIGN KEY
(
  [CTO_ServerID]
  ) REFERENCES [dbo].[CCTVServer] (
  [CSV_ID]
  ) NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVTour_CCTVSwitch] FOREIGN KEY
(
  [CTO_ParentID]
  ) REFERENCES [dbo].[CCTVSwitch] (
  [CSW_ID]
  )  NOT FOR REPLICATION,
  CONSTRAINT [FK_CCTVTour_partition] FOREIGN KEY
(
  [CTO_Partition]
  ) REFERENCES [dbo].[partition] (
  [part_number]
```

```
  )NOT FOR REPLICATION
  ) ON [PRIMARY]
  ;
```

CREATE TABLE [CCTVTour_save]

```
(
  [CTO_ID]              [int] NOT NULL,                          -- Unique ID of tour record
  [CTO_ParentID]        [int] NOT NULL,                          -- ID of parent record
  [CTO_Name]            [char] (64) NOT NULL,                    -- OPC name of tour
  [CTO_ServerID]        [int] NOT NULL,                          -- ID of server record
  [CTO_Exists]          [int] NOT NULL,                          -- 0 = No, 1 = Yes, 2 = Use default
  [CTO_Description]     [char] (64) NOT NULL,                    -- Description
  [CTO_Partition]       [int] NOT NULL,                          -- Partition ID to which this tour belongs
  [CTO_Public]          [int] NOT NULL,                          -- 1 = tour is public, 0 = tour is private
  site                  CHAR(32) NOT NULL,                       -- Site Name
  CTO_guid              uniqueidentifier NOT NULL,               -- Global unique record identifier
  CTO_checksum          INTEGER NOT NULL,                        -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,               -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                       -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                       -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                        -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,               -- Global unique record identifier
  CONSTRAINT [PK_CCTVTour_save] PRIMARY KEY CLUSTERED
(
  [CTO_ID], upd_mode desc, upd_guid
)  ON [PRIMARY],
) ON [PRIMARY]
;
```

CREATE TABLE company

```
(
  company_partition     INTEGER REFERENCES partition(part_number)
                          NOT FOR REPLICATION,                   -- partition ID to which this company belongs
  company_public        SMALLINT,                                -- 1 = company is public, 0 = company is private
  company_id            INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,  -- unique database ID
  company_name          CHAR(32) NOT NULL,                       -- name of company
  company_autoadded     SMALLINT NOT NULL,                       -- company was automatically added by Visitor
                                                                    Management
  site                  CHAR(32) NOT NULL,                       -- Site Name
  company_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  company_checksum      INTEGER NOT NULL,                        -- Record checksum
  PRIMARY KEY           ( company_partition, company_id )
);
  CREATE INDEX          company_name ON company ( company_name );
```

CREATE TABLE company_save

```
(
  company_partition     INTEGER NOT NULL,                        -- partition ID to which this company belongs
  company_public        SMALLINT,                                -- 1 = company is public, 0 = company is private
  company_id            INTEGER NOT NULL,                        -- unique database ID
  company_name          CHAR(32) NOT NULL,                       -- name of company
  company_autoadded     SMALLINT NOT NULL,                       -- company was automatically added by Visitor Management
  site                  CHAR(32) NOT NULL,                       -- Site Name
  company_guid          uniqueidentifier NOT NULL,               -- Global unique record identifier
  company_checksum      INTEGER NOT NULL,                        -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,               -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                       -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                       -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                        -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
```

```
                                newid() UNIQUE NOT NULL,                    -- Global unique record identifier
  PRIMARY KEY                ( company_partition, company_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE connections

```
(
  conn_partition              INTEGER REFERENCES partition(part_number)
                                NOT FOR REPLICATION,                        -- partition ID to which this connection belongs
  conn_public                 SMALLINT NOT NULL,                            -- 1 = connection is public, 0 = counter is private
  conn_id                     INTEGER IDENTITY NOT FOR
                                REPLICATION UNIQUE,                         -- unique database ID
  conn_name                   VARCHAR(64) NOT NULL,                         -- connection name
  conn_rras_name              VARCHAR(64) NULL,                             -- rras's connection name
  conn_ip                     CHAR(16) NOT NULL,                            -- starting IP address
  conn_ip_mask                CHAR(16) NOT NULL,                            -- IP address mask
  conn_phone                  VARCHAR(128) NULL,                            -- Phone number
  conn_remote_login_name      VARCHAR(64) NULL,                             -- Login name at panel end
  conn_remote_password        VARCHAR(64) NULL,                             -- Password at panel end
  conn_local_login_name       VARCHAR(64) NULL,                             -- Login name at host end
  conn_rras_local_login_name  VARCHAR(64) NULL,                             -- Login name at host end
  conn_local_password         VARCHAR(64) NULL,                             -- Password at host end
  conn_uses_rras              SMALLINT NOT NULL,                            -- 1 = Use an RRAS connection, 0 = Use 3rd party HW
  conn_has_rras               SMALLINT NOT NULL,                            -- 1 = RRAS connection already created, 0 = not yet
  conn_timeout                INTEGER NOT NULL,                             -- Seconds to wait for redial
  conn_no_of_panel            INTEGER,                                      -- Number of panels could pass the mask
  site                        CHAR(32) NOT NULL,                            -- Site Name
  conn_guid                   uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  conn_checksum               INTEGER NOT NULL,                             -- Record checksum
  PRIMARY KEY                 ( conn_partition, conn_id )
);
  CREATE INDEX                conn_public  ON connections ( conn_public );
  CREATE UNIQUE INDEX         conn_site_name ON connections (site, conn_name);
```

CREATE TABLE connections_save

```
(
  conn_partition              INTEGER NOT NULL,                             -- partition ID to which this connection belongs
  conn_public                 SMALLINT NOT NULL,                            -- 1 = connection is public, 0 = counter is private
  conn_id                     INTEGER NOT NULL,                             -- unique database ID
  conn_name                   VARCHAR(64) NOT NULL,                         -- connection name
  conn_rras_name              VARCHAR(64) NULL,                             -- rras's connection name
  conn_ip                     CHAR(16) NOT NULL,                            -- starting IP address
  conn_ip_mask                CHAR(16) NOT NULL,                            -- IP address mask
  conn_phone                  VARCHAR(128) NULL,                            -- Phone number
  conn_remote_login_name      VARCHAR(64) NULL,                             -- Login name at panel end
  conn_remote_password        VARCHAR(64) NULL,                             -- Password at panel end
  conn_local_login_name       VARCHAR(64) NULL,                             -- Login name at host end
  conn_rras_local_login_name  VARCHAR(64) NULL,                             -- Login name at host end
  conn_local_password         VARCHAR(64) NULL,                             -- Password at host end
  conn_uses_rras              SMALLINT NOT NULL,                            -- 1 = Use an RRAS connection, 0 = Use 3rd party HW
  conn_has_rras               SMALLINT NOT NULL,                            -- 1 = RRAS connection already created, 0 = not yet
  conn_timeout                INTEGER NOT NULL,                             -- Seconds to wait for redial
  conn_no_of_panel            INTEGER,                                      -- Number of panels could pass the mask
  site                        CHAR(32) NOT NULL,                            -- Site Name
  conn_guid                   uniqueidentifier NOT NULL,                    -- Global unique record identifier
  conn_checksum               INTEGER NOT NULL,                             -- Record checksum
  upd_guid                    uniqueidentifier NOT NULL,                    -- Guid for update operation
  upd_mode                    SMALLINT NOT NULL,                            -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp               DATETIME NOT NULL,                            -- Date / Time of the update
  upd_checksum                INTEGER NOT NULL,                             -- Checksum
  upd_rowguid                 uniqueidentifier ROWGUIDCOL DEFAULT newid()
                                UNIQUE NOT NULL,                            -- Global unique record identifier
  PRIMARY KEY                 ( conn_partition, conn_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE counter

```
(
  ctr_partition              INTEGER REFERENCES partition(part_number)
                               NOT FOR REPLICATION,                    -- partition ID to which this counter belongs
  ctr_public                 SMALLINT,                                 -- 1 = counter is public, 0 = counter is private
  ctr_id                     INTEGER IDENTITY NOT FOR REPLICATION UNIQUE, -- unique database ID
  ctr_name                   CHAR(32) NOT NULL,                        -- counter name
  ctr_value                  INTEGER,                                  -- current counter value
  site                       CHAR(32) NOT NULL,                        -- Site Name
  ctr_guid                   uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  ctr_checksum               INTEGER NOT NULL,                         -- Record checksum
  PRIMARY KEY                ( ctr_partition, ctr_id )
);
  CREATE INDEX               ctr_public  ON counter ( ctr_public );
  CREATE UNIQUE INDEX        ctr_site_name ON counter (site, ctr_name);
```

CREATE TABLE counter_p900

```
(
  cp_id                      INTEGER IDENTITY NOT FOR REPLICATION UNIQUE, -- Unique database ID of Cotag counter-- the counter ID
  cp_name                    varchar(32),                              -- the counter name
  cp_number                  SMALLINT NOT NULL,                        -- the physical counter number
  cp_status                  SMALLINT NOT NULL,                        -- the counter status
  cp_report_change           SMALLINT NOT NULL,                        -- report when change
  cp_current_value           INTEGER,                                  -- the current value for counter control
  cp_default_value           INTEGER,                                  -- the default counter value
  cp_max_value               INTEGER,                                  -- the maximum counter value
  cp_force_value             INTEGER,                                  -- the force value
  cp_update_time             DATETIME,                                 -- the time stamp for control
  cp_query_string            CHAR(64),                                 -- query string enter by user
  cp_partition               INTEGER REFERENCES partition(part_number)
                               NOT FOR REPLICATION,                    -- ID of partition for this counter
  cp_public                  SMALLINT NOT NULL,                        -- open to punlic
  cp_panel_id                INTEGER NOT NULL REFERENCES panel(p_panel_id)
                               NOT FOR REPLICATION,                    -- Panel ID
  site                       CHAR(32) NOT NULL,                        -- Site Name
  cp_guid                    uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  cp_checksum                INTEGER NOT NULL,                         -- Record checksum
  PRIMARY KEY                (cp_partition, cp_id )
);
  CREATE INDEX               cp_public ON counter_p900(cp_public);
  CREATE UNIQUE INDEX        cp_number ON counter_p900 (cp_number, cp_panel_id);
  CREATE UNIQUE INDEX        cp_site_name ON counter_p900 (site, cp_name);
```

CREATE TABLE counter_p900_save

```
(
  cp_id                      INTEGER,                                  -- Unique database ID of Cotag counter-- the counter ID
  cp_name                    varchar(32),                              -- the counter name
  cp_number                  SMALLINT NOT NULL,                        -- the physical counter number
  cp_status                  SMALLINT NOT NULL,                        -- the counter status
  cp_report_change           SMALLINT NOT NULL,                        -- report when change
  cp_current_value           INTEGER,                                  -- the current value for counter control
  cp_default_value           INTEGER,                                  -- the default counter value
  cp_max_value               INTEGER,                                  -- the maximum counter value
  cp_force_value             INTEGER,                                  -- the force value
  cp_update_time             DATETIME,                                 -- the time stamp for control
  cp_query_string            CHAR(64),                                 -- query string enter by user
  cp_partition               INTEGER NOT NULL,                         -- ID of partition for this counter
  cp_public                  SMALLINT NOT NULL,                        -- open to punlic
  cp_panel_id                INTEGER NOT NULL,                         -- Panel ID
  site                       CHAR(32) NOT NULL,                        -- Site Name
  cp_guid                    uniqueidentifier NOT NULL,                -- Global unique record identifier
  cp_checksum                INTEGER NOT NULL,                         -- Record checksum
```

```
  upd_guid          uniqueidentifier NOT NULL,                    -- Guid for update operation
  upd_mode          SMALLINT NOT NULL,                            -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp     DATETIME NOT NULL,                            -- Date / Time of the update
  upd_checksum      INTEGER NOT NULL,                             -- Checksum
  upd_rowguid       uniqueidentifier ROWGUIDCOL DEFAULT newid()
                       UNIQUE NOT NULL,                           -- Global unique record identifier
  PRIMARY KEY       (cp_partition, cp_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE counter_save

```
(
  ctr_partition     INTEGER not null,                             -- partition ID to which this counter belongs
  ctr_public        SMALLINT,                                     -- 1 = counter is public, 0 = counter is private
  ctr_id            INTEGER NOT NULL,                             -- unique database ID
  ctr_name          CHAR(32) NOT NULL,                            -- counter name
  ctr_value         INTEGER,                                      -- current counter value
  site              CHAR(32) NOT NULL,                            -- Site Name
  ctr_guid          uniqueidentifier NOT NULL,                    -- Global unique record identifier
  ctr_checksum      INTEGER NOT NULL,                             -- Record checksum
  upd_guid          uniqueidentifier NOT NULL,                    -- Guid for update operation
  upd_mode          SMALLINT NOT NULL,                            -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp     DATETIME NOT NULL,                            -- Date / Time of the update
  upd_checksum      INTEGER NOT NULL,                             -- Checksum
  upd_rowguid       uniqueidentifier ROWGUIDCOL DEFAULT newid()
                       UNIQUE NOT NULL,                           -- Global unique record identifier
  PRIMARY KEY       ( ctr_partition, ctr_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE dataimport_fields

```
(
  dif_id            INTEGER IDENTITY NOT FOR REPLICATION PRIMARY KEY,   -- unique database ID
  dif_interface     CHAR(32) NOT NULL,                                  -- interface name ref by p2000
  dif_db            CHAR(32) NOT NULL,                                  -- database
  dif_table         CHAR(128) NOT NULL,                                 -- database table name
  dif_field         CHAR(64) NOT NULL,                                  -- table field name
  site              CHAR(32) NOT NULL,                                  -- Site Name
  dif_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  dif_checksum      INTEGER NOT NULL,                                   -- Record checksum
);
  CREATE INDEX      dif_interface ON dataimport_fields ( dif_interface );
```

CREATE TABLE dataimport_fields_save

```
(
  dif_id            INTEGER NOT NULL,                             -- unique database ID
  dif_interface     CHAR(32) NOT NULL,                            -- interface name ref by p2000
  dif_db            CHAR(32) NOT NULL,                            -- database
  dif_table         CHAR(128) NOT NULL,                           -- database table name
  dif_field         CHAR(64) NOT NULL,                            -- table field name
  site              CHAR(32) NOT NULL,                            -- Site Name
  dif_guid          uniqueidentifier NOT NULL,                    -- Global unique record identifier
  dif_checksum      INTEGER NOT NULL,                             -- Record checksum
  upd_guid          uniqueidentifier NOT NULL,                    -- Guid for update operation
  upd_mode          SMALLINT NOT NULL,                            -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp     DATETIME NOT NULL,                            -- Date / Time of the update
  upd_checksum      INTEGER NOT NULL,                             -- Checksum
  upd_rowguid       uniqueidentifier ROWGUIDCOL DEFAULT newid()
                       UNIQUE NOT NULL,                           -- Global unique record identifier
  PRIMARY KEY       (dif_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE dataimport_mapping

```
(
  dim_id            INTEGER IDENTITY NOT FOR REPLICATION PRIMARY KEY,   -- unique database ID
  dim_interface     CHAR(32) NOT NULL,                                  -- interface name ref by p2000
  dim_sourcedb      CHAR(32) NOT NULL,                                  -- source db
  dim_sourcetable   CHAR(128) NOT NULL,                                 -- source db table name
  dim_sourcefield   CHAR(64) NOT NULL,                                  -- source db field
  dim_savedb        CHAR(32) NOT NULL,                                  -- db where source info will be saved
  dim_savetable     CHAR(128) NOT NULL,                                 -- destination db table name
  dim_savefield     CHAR(64) NOT NULL,                                  -- destination db field
  site              CHAR(32) NOT NULL,                                  -- Site Name
  dim_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  dim_checksum      INTEGER NOT NULL,                                   -- Record checksum
);
  CREATE INDEX      dim_interface ON dataimport_mapping ( dim_interface );
```

CREATE TABLE dataimport_mapping_save

```
(
  dim_id            INTEGER,                                            -- unique database ID
  dim_interface     CHAR(32) NOT NULL,                                  -- interface name ref by p2000
  dim_sourcedb      CHAR(32) NOT NULL,                                  -- source db
  dim_sourcetable   CHAR(128) NOT NULL,                                 -- source db table name
  dim_sourcefield   CHAR(64) NOT NULL,                                  -- source db field
  dim_savedb        CHAR(32) NOT NULL,                                  -- db where source info will be saved
  dim_savetable     CHAR(128) NOT NULL,                                 -- destination db table name
  dim_savefield     CHAR(64) NOT NULL,                                  -- destination db field
  site              CHAR(32) NOT NULL,                                  -- Site Name
  dim_guid          uniqueidentifier NOT NULL,                          -- Global unique record identifier
  dim_checksum      INTEGER NOT NULL,                                   -- Record checksum
  upd_guid          uniqueidentifier NOT NULL,                          -- Guid for update operation
  upd_mode          SMALLINT NOT NULL,                                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp     DATETIME NOT NULL,                                  -- Date / Time of the update
  upd_checksum      INTEGER NOT NULL,                                   -- Checksum
  upd_rowguid       uniqueidentifier ROWGUIDCOL DEFAULT newid()
                      UNIQUE NOT NULL,                                  -- Global unique record identifier
  PRIMARY KEY       (dim_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE dbversion

```
(
  major             INTEGER NOT NULL,                   -- database major version
  minor             INTEGER NOT NULL,                   -- database minor version
  release           INTEGER NOT NULL,                   -- database release number
  build             INTEGER NOT NULL,                   -- database build number
  installdate       DATETIME NOT NULL                   -- date when this build was installed
);
```

CREATE TABLE dept

```
(
  dept_partition    INTEGER REFERENCES partition(part_number)
                      NOT FOR REPLICATION,              -- partition ID to which this department belongs
  dept_public       SMALLINT,                           -- 1 = department is public, 0 = department is private
  dept_id           INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,  -- unique database ID
  dept_name         CHAR(32) NOT NULL,                  -- name of department
  site              CHAR(32) NOT NULL,                  -- Site Name
  dept_guid         uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  dept_checksum     INTEGER NOT NULL,                   -- Record checksum
  PRIMARY KEY       ( dept_partition, dept_id )
);
  CREATE INDEX      dept_name ON dept ( dept_name );
```

CREATE TABLE dept_save

```
(
  dept_partition        INTEGER NOT NULL,                          -- partition ID to which this department belongs
  dept_public           SMALLINT,                                  -- 1 = department is public, 0 = department is private
  dept_id               INTEGER NOT NULL,                          -- unique database ID
  dept_name             CHAR(32) NOT NULL,                         -- name of department
  site                  CHAR(32) NOT NULL,                         -- Site Name
  dept_guid             uniqueidentifier NOT NULL,                 -- Global unique record identifier
  dept_checksum         INTEGER NOT NULL,                          -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                 -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                         -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                          -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,                 -- Global unique record identifier
  PRIMARY KEY           ( dept_partition, dept_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE dnld_req_rule

```
(
  dr_type               smallint NOT NULL,                         -- request type
  dr_entry_time         datetime NULL,                             -- the entry time of the request
  dr_scheduled_time     datetime NULL,                             -- the entry time of the request
  dr_workstation        varchar (64) NULL,                         -- User name to make the request
  dr_user_name          varchar (32) NULL,                         -- User name to make the request
  dr_status             smallint NOT NULL,                         -- the current status, Waiting = 0, Queued = 1, Processing = 3
  dr_lparam1            int NULL,                                  -- 1st data identifier Int
  dr_szparam1           varchar (64) NULL,                         -- 1st data identifier String
  dr_lparam2            int NULL,                                  -- 2nd data identifier Int
  dr_szparam2           varchar (64) NULL,                         -- 2st data identifier String
  site                  CHAR(32) NOT NULL,                         -- Site Name
  dr_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY           NONCLUSTERED (dr_guid)
)
  CREATE INDEX          dr_type ON dnld_req_rule (dr_type);
  CREATE INDEX          dnld_req_rule_site ON dnld_req_rule(site);
  CREATE INDEX          dnld_req_rule_scheduled ON dnld_req_rule(dr_scheduled_time);
```

CREATE TABLE dnld_req_timed

```
(
  dr_type               smallint NOT NULL,                         -- request type
  dr_entry_time         datetime NULL,                             -- the entry time of the request
  dr_scheduled_time     datetime NULL,                             -- the entry time of the request
  dr_workstation        varchar (64) NULL,                         -- User name to make the request
  dr_user_name          varchar (32) NULL,                         -- User name to make the request
  dr_status             smallint NOT NULL,                         -- the current status, Waiting = 0, Queued = 1, Processing = 3
  dr_lparam1            int NULL,                                  -- 1st data identifier
  dr_szparam1           varchar (64) NULL,                         -- 1st data identifier String
  dr_lparam2            int NULL,                                  -- 2nd data identifier
  dr_szparam2           varchar (64) NULL,                         -- 2st data identifier String
  site                  CHAR(32) NOT NULL,                         -- Site Name
  dr_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY           NONCLUSTERED (dr_guid)
)
  CREATE INDEX          dr_type ON dnld_req_timed (dr_type);
  CREATE INDEX          dnld_req_timed_site ON dnld_req_timed(site);
  CREATE INDEX          dnld_req_timed_scheduled ON dnld_req_rule(dr_scheduled_time);
```

CREATE TABLE drycontact_status

```
(
  dcstat_id            INTEGER PRIMARY KEY REFERENCES AVDryContact(ADC_ID)
                         NOT FOR REPLICATION,                          -- dry contact ID to which this status belongs
  dcstat_status        SMALLINT NOT NULL,                              -- current state for dry contact
  dcstat_timestamp     DATETIME NOT NULL,                              -- time of last current dry contact
  site                 CHAR(32) NOT NULL,                              -- Site Name
  dcstat_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
  dcstat_checksum      INTEGER NOT NULL,                               -- Record checksum
);
```

CREATE TABLE elevator

```
(
  el_partition         INTEGER REFERENCES partition(part_number)
                         NOT FOR REPLICATION,                          -- partition ID to which this elevator/cabinet belongs to
  el_public            SMALLINT,                                       -- 1 = public, 0 = private
  el_elevator_id       INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,-- unique elevator/cabinet ID
  el_elevator_name     CHAR(32) NOT NULL,                              -- elevator/cabinet name
  el_type              SMALLINT,                                       -- item type, 1 = elevator, 2 = cabinet
  el_panel_id          INTEGER REFERENCES panel(p_panel_id)
                         NOT FOR REPLICATION,                          -- panel ID to which this elevator/cabinet belongs to
  el_if_level          SMALLINT,                                       -- 1 = high level I/F, 0 = low level I/F
  el_high_protocol     INTEGER,                                        -- TBD
  el_timed_access      SMALLINT,                                       -- 0 = no timed access, 1 = timed access
  el_floor_track       SMALLINT,                                       -- 0 = no floor/door tracking, 1 = floor/door tracking
  el_access_time       INTEGER,                                        -- access time (sec's)
  el_cab_reader        INTEGER REFERENCES terminal (tp_term_id)
                         NOT FOR REPLICATION,                          -- reader ID
  el_fireman_sw        INTEGER NULL REFERENCES input(ip_point_id)
                         NOT FOR REPLICATION,                          -- fireman/emergency switch input
  el_service_sw        INTEGER NULL REFERENCES input(ip_point_id)
                         NOT FOR REPLICATION,                          -- service switch input
  el_suppress_time     INTEGER,                                        -- door in alarm supp time
  el_alarm_flag        SMALLINT,                                       -- door input alarm flags,
                                                                       -- 0 = no input alarms, 1 = input alarm enabled
  el_query_string      char(64) NULL,
  el_track_open        SMALLINT,                                       -- 1 = track on input open
  el_track_transition  SMALLINT,                                       -- 1 = track on input transition
  el_HLEI_user_var     SMALLINT NOT NULL,                              -- for KONE, elevator address (1 to 8),
                                                                          Kone Ip Elevator addr or Level
  site                 CHAR(32) NOT NULL,                              -- Site Name
  el_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
  el_checksum          INTEGER NOT NULL,                               -- Record checksum
  el_d620_ecg_mode     SMALLINT NOT NULL,
  el_device_address    SMALLINT NOT NULL,                              -- for Otis Compass, address of (MRE * 256) + DEC,
                                                                          Kone Ip Group Address
  el_ada_special_flag  SMALLINT NOT NULL,                              -- 0 = none 1 - 3 = special access flags a - c
  el_vip_special_flag  SMALLINT NOT NULL,                              -- 0 = none 1 - 3 = special access flags a - c
  el_pin_allowed       SMALLINT NOT NULL,                              -- 1 = allow pin code entry on otis compass
  el_sub_type          SMALLINT NOT NULL,                              -- 0 = not used, 1 = Kone IP Dop or 2 = Kone IP Cop
  PRIMARY KEY          (el_partition, el_elevator_id)
);
  CREATE INDEX         el_public ON elevator (el_public);
  CREATE INDEX         el_type  ON elevator (el_type);
  CREATE INDEX         el_panel_id ON elevator (el_panel_id);
  CREATE UNIQUE INDEX  el_elevator_name_type ON elevator (site, el_elevator_name, el_type);
```

CREATE TABLE elevatordetails

```
(
  ed_elevator_id       INTEGER REFERENCES elevator (el_elevator_id) NOT FOR REPLICATION,   -- elevator/cabinet ID
  ed_floor_number      SMALLINT,                                                          -- floor/door number for this
                                                                                             elevator/cabinet
```

```
  ed_output          INTEGER NULL REFERENCES output(op_output_id) NOT FOR REPLICATION,   -- floor/door call enable
  ed_input           INTEGER NULL REFERENCES input(ip_point_id) NOT FOR REPLICATION,     -- floor/door call monitor
  ed_timezone        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- public access timezone id
  site               CHAR(32) NOT NULL,                                                  -- Site Name
  ed_guid            uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                        -- Global unique record identifier
  ed_checksum        INTEGER NOT NULL,                                                   -- Record checksum
  ed_behavior        SMALLINT NOT NULL,                                                  -- Added for Kone Ip
  PRIMARY KEY        (ed_elevator_id, ed_floor_number)
);
```

CREATE TABLE elevatordetails_save

```
(
  ed_elevator_id     INTEGER,                                                            -- elevator/cabinet ID
  ed_floor_number    SMALLINT,                                                           -- floor/door number for this elevator/cabinet
  ed_output          INTEGER NULL,                                                       -- floor/door call enable
  ed_input           INTEGER NULL,                                                       -- floor/door call monitor
  ed_timezone        INTEGER NULL,                                                       -- public access timezone id
  site               CHAR(32) NOT NULL,                                                  -- Site Name
  ed_guid            uniqueidentifier NOT NULL,                                          -- Global unique record identifier
  ed_checksum        INTEGER NOT NULL,                                                   -- Record checksum
  upd_guid           uniqueidentifier NOT NULL,                                          -- Guid for update operation
  upd_mode           SMALLINT NOT NULL,                                                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp      DATETIME NOT NULL,                                                  -- Date / Time of the update
  upd_checksum       INTEGER NOT NULL,                                                   -- Checksum
  upd_rowguid        uniqueidentifier ROWGUIDCOL DEFAULT newid()
                        UNIQUE NOT NULL,                                                 -- Global unique record identifier
  ed_behavior        SMALLINT,                                                           -- Added for Kone Ip
  PRIMARY KEY        (ed_elevator_id, ed_floor_number, upd_mode desc, upd_guid)
);
```

CREATE TABLE elevator_save

```
(
  el_partition       INTEGER NOT NULL,                 -- partition ID to which this elevator/cabinet belongs to
  el_public          SMALLINT,                         -- 1 = public, 0 = private
  el_elevator_id     INTEGER NOT NULL,                 -- unique elevator/cabinet ID
  el_elevator_name   CHAR(32) NOT NULL,                -- elevator/cabinet name
  el_type            SMALLINT,                         -- item type, 1 = elevator, 2 = cabinet
  el_panel_id        INTEGER,                          -- panel ID to which this elevator/cabinet belongs to
  el_if_level        SMALLINT,                         -- 1 = high level I/F, 0 = low level I/F
  el_high_protocol   INTEGER,                          -- TBD
  el_timed_access    SMALLINT,                         -- 0 = no timed access, 1 = timed access
  el_floor_track     SMALLINT,                         -- 0 = no floor/door tracking, 1 = floor/door tracking
  el_access_time     INTEGER,                          -- access time (sec's)
  el_cab_reader      INTEGER,                          -- reader ID
  el_fireman_sw      INTEGER NULL,                     -- fireman/emergency switch input
  el_service_sw      INTEGER NULL,                     -- service switch input
  el_suppress_time   INTEGER,                          -- door in alarm supp time
  el_alarm_flag      SMALLINT,                         -- door input alarm flags,
                                                       -- 0 = no input alarms, 1 = input alarm enabled
  el_query_string    char(64) NULL,
  el_track_open      SMALLINT,                         -- 1 = track on input open
  el_track_transition SMALLINT,                        -- 1 = track on input transition
  el_HLEI_user_var   SMALLINT NOT NULL,                -- for KONE, elevator address (1 to 8)
  site               CHAR(32) NOT NULL,                -- Site Name
  el_guid            uniqueidentifier NOT NULL,        -- Global unique record identifier
  el_checksum        INTEGER NOT NULL,                 -- Record checksum
  upd_guid           uniqueidentifier NOT NULL,        -- Guid for update operation
  upd_mode           SMALLINT NOT NULL,                -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp      DATETIME NOT NULL,                -- Date / Time of the update
  upd_checksum       INTEGER NOT NULL,                 -- Checksum
  upd_rowguid        uniqueidentifier ROWGUIDCOL DEFAULT newid()
                        UNIQUE NOT NULL,               -- Global unique record identifier
```

```
  el_d620_ecg_mode      SMALLINT,
  el_device_address     SMALLINT,                    -- for Otis Compass, address of (MRE * 256) + DEC
  el_ada_special_flag   SMALLINT,                    -- 0 = none 1 - 3 = special access flags a - c
  el_vip_special_flag   SMALLINT,                    -- 0 = none 1 - 3 = special access flags a - c
  el_pin_allowed        SMALLINT,                    -- 1 = allow pin code entry on otis compass
  el_sub_type           SMALLINT,                    -- 0 = not used, 1 = Kone IP Dop or 2 = Kone IP Cop
  PRIMARY KEY           (el_partition, el_elevator_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE emailqueue

```
(
  email_id              INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,   -- unique database ID
  email_to              VARCHAR(512) NOT NULL,
  email_cc              VARCHAR(512) NULL,
  email_subject         VARCHAR(512) NOT NULL,
  email_body            TEXT NOT NULL,
  email_retry_count     INT NOT NULL DEFAULT 0,
  email_send_time       DATETIME NULL,
  PRIMARY KEY           ( email_id )
);
```

CREATE TABLE enable_code

```
(
  ec_id                 INTEGER IDENTITY NOT FOR REPLICATION PRIMARY KEY,  --unique db id
  ec_panel_id           INTEGER REFERENCES panel (p_panel_id)
                          NOT FOR REPLICATION,                             -- ID of panel for this enable code
  ec_code_number1       SMALLINT,                                        --enable code # 1 (0=none, 1=Ext timed override)
  ec_cn1_digit          INTEGER,                                         --enable code digit 1
  ec_code_number2       SMALLINT,                                        --enable code # 2 (0=none, 2=ext shunt time)
  ec_cn2_digit          INTEGER,                                         --enable code digit 2
  ec_code_number3       SMALLINT,                                        --enable code # 3 (0=none, 3=air crew pin)
  ec_cn3_digit          INTEGER,                                         --enable code digit 3
  ec_code_number4       SMALLINT,                                        --enable code # 4 (0=none, 4=pin+1)
  ec_cn4_digit          INTEGER,                                         --enable code digit 4
  site                  CHAR(32) NOT NULL,                               -- Site Name
  ec_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,     -- Global unique record identifier
  ec_checksum           INTEGER NOT NULL,                                -- Record checksum
);
```

CREATE TABLE enable_code_save

```
(
  ec_id                 INTEGER NOT NULL,                                --unique db id
  ec_panel_id           INTEGER,                                         -- ID of panel for this enable code
  ec_code_number1       SMALLINT,                                        --enable code # 1 (0=none, 1=Ext timed override)
  ec_cn1_digit          INTEGER,                                         --enable code digit 1
  ec_code_number2       SMALLINT,                                        --enable code # 2 (0=none, 2=ext shunt time)
  ec_cn2_digit          INTEGER,                                         --enable code digit 2
  ec_code_number3       SMALLINT,                                        --enable code # 3 (0=none, 3=air crew pin)
  ec_cn3_digit          INTEGER,                                         --enable code digit 3
  ec_code_number4       SMALLINT,                                        --enable code # 4 (0=none, 4=pin+1)
  ec_cn4_digit          INTEGER,                                         --enable code digit 4
  site                  CHAR(32) NOT NULL,                               -- Site Name
  ec_guid               uniqueidentifier NOT NULL,                       -- Global unique record identifier
  ec_checksum           INTEGER NOT NULL,                                -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                       -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                               -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                               -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                               -- Global unique record identifier
  PRIMARY KEY           (ec_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE enterprise_parameters

```
(
  enp_name              char(32) not null,                                  -- Name of main site within enterprise
  enp_name_alt          char(32),                                           -- Name of alternate publisher site within enterprise
  site                  CHAR(32) NOT NULL,                                  -- Site Name
  enp_guid              uniqueidentifier ROWGUIDCOL PRIMARY KEY NOT NULL,   -- Global unique record identifier
  enp_checksum          INTEGER NOT NULL,                                   -- Record Digital Signature
);
```

CREATE TABLE enterprise_parameters_save

```
(
  enp_name              char(32) not null,                                  -- Name of main site within enterprise
  enp_name_alt          char(32),                                           -- Name of alternate publisher site within enterprise
  site                  CHAR(32) NOT NULL,                                  -- Site Name
  enp_guid              uniqueidentifier NOT NULL,                          -- Global unique record identifier
  enp_checksum          INTEGER NOT NULL,                                   -- Record Digital Signature
  upd_guid              uniqueidentifier NOT NULL,                          -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                  -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                   -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                  -- Global unique record identifier
  PRIMARY KEY           (enp_guid, upd_mode desc, upd_guid)
);
```

CREATE TABLE ent_site

```
(
  ens_id                INTEGER IDENTITY NOT FOR REPLICATION PRIMARY KEY,   --unique db id
  ens_name              CHAR(32) UNIQUE NOT NULL,                           -- Enterprise Site Name
  ens_host_name         CHAR(32) NOT NULL,                                  -- Enterprise Site DB Server Name
  ens_host_name_alt     CHAR(32),                                           -- Alternate Enterprise Site DB Server Name
  ens_subscribe_to_all  SMALLINT NOT NULL,                                  -- 0-No, 1-Yes.
  site                  CHAR(32) NOT NULL,                                  -- Site Name
  ens_guid              uniqueidentifier UNIQUE ROWGUIDCOL NOT NULL,        -- Global unique record identifier
  ens_checksum          INTEGER NOT NULL,                                   -- Record checksum
);
```

CREATE TABLE esp_parameters

```
(
  esp_guid                  uniqueidentifier ROWGUIDCOL PRIMARY KEY,                --Global unique record identifier
  esp_ens_guid_subscriber   uniqueidentifier NOT NULL REFERENCES ent_site(ens_guid)
                              NOT FOR REPLICATION,                                  -- Guid of the subscriber's site name
  esp_ens_guid_guid         uniqueidentifier NOT NULL REFERENCES ent_site(ens_guid)
                              NOT FOR REPLICATION,                                  -- Guid of the enterprise site
  esp_enabled               SMALLINT NOT NULL,                                      -- 1 if enabled, 0 if not enabled
  site                      CHAR(32) NOT NULL,                                      -- Site Name,
  esp_checksum              INTEGER NOT NULL,                                       -- Record checksum
);
  CREATE UNIQUE INDEX   esp_ens_guid_subscriber ON esp_parameters ( esp_ens_guid_subscriber, esp_ens_guid_guid);
```

CREATE TABLE esp_parameters_save

```
(
  esp_guid                  uniqueidentifier,                                       --Global unique record identifier
  esp_ens_guid_subscriber   uniqueidentifier NOT NULL,                              -- Guid of the subscriber's site name
  esp_ens_guid_guid         uniqueidentifier NOT NULL,                              -- Guid of the enterprise site
  esp_enabled               SMALLINT NOT NULL,                                      -- 1 if enabled, 0 if not enabled
  site                      CHAR(32) NOT NULL,                                      -- Site Name,
  esp_checksum              INTEGER NOT NULL,                                       -- Record checksum
  upd_guid                  uniqueidentifier NOT NULL,                              -- Guid for update operation
  upd_mode                  SMALLINT NOT NULL,                                      -- Update mode (0 - delete, 1 - edit, 2 - new)
```

```
  upd_timestamp          DATETIME NOT NULL,                                          -- Date / Time of the update
  upd_checksum           INTEGER NOT NULL,                                           -- Checksum
  upd_rowguid            uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY            (esp_guid, upd_mode desc, upd_guid)
);
```

CREATE TABLE ent_site_save

```
(
  ens_id                 INTEGER,                                    --unique db id
  ens_name               CHAR(32),                                   -- Enterprise Site Name
  site                   CHAR(32) NOT NULL,                          -- Site Name
  ens_host_name          CHAR(32) NOT NULL,                          -- Enterprise Site DB Server Name
  ens_host_name_alt      CHAR(32),                                   -- Alternate Enterprise Site DB Server Name
  ens_subscribe_to_all   SMALLINT NOT NULL,                          -- 0-No, 1-Yes.
  ens_guid               uniqueidentifier NOT NULL,                  -- Global unique record identifier
  ens_checksum           INTEGER NOT NULL,                           -- Record checksum
  upd_guid               uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode               SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp          DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum           INTEGER NOT NULL,                           -- Checksum
  upd_rowguid            uniqueidentifier ROWGUIDCOL DEFAULT newid()
                           UNIQUE NOT NULL,                          -- Global unique record identifier
  PRIMARY KEY            (ens_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE event

```
(
  ev_partition           INTEGER REFERENCES partition(part_number)
                           NOT FOR REPLICATION,                      -- partition ID to which this event belongs
  ev_public              SMALLINT,                                   -- 1 = public, 0 = private
  ev_id                  INTEGER IDENTITY NOT FOR REPLICATION UNIQUE, -- unique database ID for this event
  ev_name                CHAR(32) NOT NULL,                          -- event name
  ev_manual_activate     SMALLINT,                                   -- 1 = allow manual activation, 0 = disallow
  ev_active_timezone     INTEGER NULL REFERENCES timezone(tz_id)
                           NOT FOR REPLICATION,                      -- timezone ID when this event is active, if
                                                                       NULL then always active
  ev_logic               SMALLINT,                                   -- logic for triggers (0 = AND, 1 = OR)
  ev_last_activate       DATETIME,                                   -- last date/time event was activated
  site                   CHAR(32) NOT NULL,                          -- Site Name
  ev_guid                uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  ev_checksum            INTEGER NOT NULL,                           -- Record checksum
  ev_enable              SMALLINT NOT NULL,
  PRIMARY KEY            ( ev_partition, ev_id )
);
  CREATE INDEX           ev_public ON event(ev_public);
  CREATE UNIQUE INDEX    ev_site_name ON event(site, ev_name);
  CREATE INDEX           ev_enable ON event(ev_enable);
```

CREATE TABLE event_action

```
(
  eva_id                 INTEGER IDENTITY NOT FOR REPLICATION
                           PRIMARY KEY,                              -- unique id of this action
  eva_event_id           INTEGER REFERENCES event(ev_id)
                           NOT FOR REPLICATION,                      -- ID of event for this action
  eva_order              SMALLINT,                                   -- order of this action in its event
  eva_delay              INTEGER,                                    -- seconds to delay before executing this action
  eva_type               SMALLINT,                                   -- action type (see eventactiondb.h)
  eva_value1             INTEGER NULL,                               -- value 1, usage depends upon eva_type (see eventactiondb.h)
  eva_value2             INTEGER NULL,                               -- value 2, usage depends upon eva_type (see eventactiondb.h)
  eva_partition          INTEGER NULL REFERENCES partition(part_number)
                           NOT FOR REPLICATION,                      -- ID of partition for this action
```

```
  eva_text              CHAR(255) NULL,                          -- value text, usage depends upon eva_type
                                                                    (see eventactiondb.h)
  eva_value_float       FLOAT NULL,                              -- value 3, usage depends upon eva_type (see eventactiondb.h)
  eva_value_str         CHAR(255) NULL,                          -- value 4, usage depends upon eva_type (see eventactiondb.h)
  site                  CHAR(32) NOT NULL,                       -- Site Name
  eva_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  eva_checksum          INTEGER NOT NULL,                        -- Record checksum
  eva_value3            INTEGER NULL,
);
  CREATE INDEX          eva_event_id ON event_action ( eva_event_id );
  CREATE INDEX          eva_order ON event_action ( eva_order );
  CREATE INDEX          eva_type ON event_action ( eva_type );
```

CREATE TABLE event_action_decode

```
(
  evad_id               INTEGER REFERENCES event_action(eva_id) NOT
                          FOR REPLICATION PRIMARY KEY,           -- unique id of this action
  evad_event_id         INTEGER REFERENCES event(ev_id) NOT FOR REPLICATION, -- ID of event for this action
  evad_order            SMALLINT,                                -- order of this action in its event
  evad_delay            INTEGER,                                 -- seconds to delay before executing this action
  evad_category         VARCHAR(256),
  evad_type             SMALLINT,                                -- action type (see eventactiondb.h)
  evad_type_text        VARCHAR(256),
  evad_value1_text      VARCHAR(256),
  evad_value1_value     VARCHAR(256),
  evad_value2_text      VARCHAR(256),
  evad_value2_value     VARCHAR(256),
  evad_value3_text      VARCHAR(256),
  evad_value3_value     VARCHAR(256),
  site                  CHAR(32) NOT NULL,                       -- Site Name
  evad_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  evad_checksum         INTEGER NOT NULL,                        -- Record checksum
);
  CREATE INDEX          evad_event_id ON event_action_decode ( evad_event_id );
  CREATE INDEX          evad_order ON event_action_decode ( evad_order );
```

CREATE TABLE event_action_save

```
(
  eva_id                INTEGER NOT NULL,                        -- unique id of this action
  eva_event_id          INTEGER,                                 -- ID of event for this action
  eva_order             SMALLINT,                                -- order of this action in its event
  eva_delay             INTEGER,                                 -- seconds to delay before executing this action
  eva_type              SMALLINT,                                -- action type (see eventactiondb.h)
  eva_value1            INTEGER NULL,                            -- value 1, usage depends upon eva_type (see eventactiondb.h)
  eva_value2            INTEGER NULL,                            -- value 2, usage depends upon eva_type (see eventactiondb.h)
  eva_partition         INTEGER NULL,                            -- ID of partition for this action
  eva_text              CHAR(255) NULL,                          -- value text, usage depends upon eva_type
                                                                    (see eventactiondb.h)
  eva_value_float       FLOAT NULL,                              -- value 3, usage depends upon eva_type (see eventactiondb.h)
  eva_value_str         CHAR(255) NULL,                          -- value 4, usage depends upon eva_type (see eventactiondb.h)
  site                  CHAR(32) NOT NULL,                       -- Site Name
  eva_guid              uniqueidentifier NOT NULL,               -- Global unique record identifier
  eva_checksum          INTEGER NOT NULL,                        -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,               -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                       -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                       -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                        -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                       -- Global unique record identifier
  eva_value3            INTEGER NULL,
  PRIMARY KEY           (eva_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE event_trigger

```
(
  evt_id                INTEGER IDENTITY NOT FOR REPLICATION
                          PRIMARY KEY,                          -- unique ID for this trigger
  evt_event_id          INTEGER REFERENCES event(ev_id)
                          NOT FOR REPLICATION,                  -- ID of event for this trigger
  evt_type              INTEGER,                                -- trigger type (see eventtriggerdb.h)
  evt_condition_type    SMALLINT NULL,                          -- trigger condition type (see eventtriggerdb.h)
  evt_condition_logic   SMALLINT NULL,                          -- trigger condition logic (see eventtriggerdb.h)
  evt_condition_value   INTEGER NULL,                           -- trigger condition value (see eventtriggerdb.h for usage)
  evt_condition_badge   VARCHAR(100) NULL REFERENCES badge(b_number_str)
                          NOT FOR REPLICATION,                  -- trigger condition badge value (if used)
  evt_condition_date    DATETIME,                               -- trigger condition date/time value (if used)
  evt_partition         INTEGER REFERENCES partition(part_number)
                          NOT FOR REPLICATION,                  -- ID of partition for this trigger
  evt_not_for_drill     SMALLINT NOT NULL,                      -- For events triggering on muster start/stop
  evt_condition_string  CHAR(255) NULL,                         -- trigger condition string
                                                                -- 1 = do not trigger for a drill, 0 always trigger
  site                  CHAR(32) NOT NULL,                      -- Site Name
  evt_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  evt_checksum          INTEGER NOT NULL,                       -- Record checksum
);
  CREATE INDEX          evt_event_id ON event_trigger ( evt_event_id );
  CREATE INDEX          evt_type ON event_trigger ( evt_type );
```

CREATE TABLE event_trigger_decode

```
(
  evtd_id               INTEGER REFERENCES event_trigger(evt_id)
                          NOT FOR REPLICATION PRIMARY KEY,      -- unique ID for this trigger
  evtd_event_id         INTEGER REFERENCES event(ev_id) NOT FOR REPLICATION,  -- ID of event for this trigger
  evtd_category         VARCHAR(256),
  evtd_type             INTEGER,                                -- trigger type (see eventtriggerdb.h)
  evtd_type_text        VARCHAR(256),
  evtd_condition_type_text  VARCHAR(256),
  evtd_condition_logic_text VARCHAR(256),
  evtd_condition_value_text VARCHAR(256),
  evtd_not_for_drill    SMALLINT NOT NULL,                      -- 1 = do not trigger for a drill, 0 always trigger
  site                  CHAR(32) NOT NULL,                      -- Site Name
  evtd_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  evtd_checksum         INTEGER NOT NULL,                       -- Record checksum
);
  CREATE INDEX          evtd_event_id ON event_trigger_decode ( evtd_event_id );
  CREATE INDEX          evtd_type ON event_trigger_decode ( evtd_type );
```

CREATE TABLE event_trigger_save

```
(
  evt_id                INTEGER NOT NULL,                       -- unique ID for this trigger
  evt_event_id          INTEGER,                                -- ID of event for this trigger
  evt_type              INTEGER,                                -- trigger type (see eventtriggerdb.h)
  evt_condition_type    SMALLINT NULL,                          -- trigger condition type (see eventtriggerdb.h)
  evt_condition_logic   SMALLINT NULL,                          -- trigger condition logic (see eventtriggerdb.h)
  evt_condition_value   INTEGER NULL,                           -- trigger condition value (see eventtriggerdb.h for usage)
  evt_condition_badge   VARCHAR(100) NULL,                      -- trigger condition badge value (if used)
  evt_condition_date    DATETIME,                               -- trigger condition date/time value (if used)
  evt_partition         INTEGER,                                -- ID of partition for this trigger
  evt_not_for_drill     SMALLINT NOT NULL,                      -- For events triggering on muster start/stop
                                                                -- 1 = do not trigger for a drill, 0 always trigger
  evt_condition_string  CHAR(255) NULL,                         -- trigger condition string
  site                  CHAR(32) NOT NULL,                      -- Site Name
  evt_guid              uniqueidentifier NOT NULL,              -- Global unique record identifier
  evt_checksum          INTEGER NOT NULL,                       -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,              -- Guid for update operation
```

```
  upd_mode              SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                           -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                          -- Global unique record identifier
  PRIMARY KEY           (evt_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE event_save

```
(
  ev_partition          INTEGER NOT NULL,                           -- partition ID to which this event belongs
  ev_public             SMALLINT,                                   -- 1 = public, 0 = private
  ev_id                 INTEGER NOT NULL,                           -- unique database ID for this event
  ev_name               CHAR(32)  NOT NULL,                         -- event name
  ev_manual_activate    SMALLINT,                                   -- 1 = allow manual activation, 0 = disallow
  ev_active_timezone    INTEGER NULL,                               -- timezone ID when this event is active, if NULL then always active
  ev_logic              SMALLINT,                                   -- logic for triggers (0 = AND, 1 = OR)
  ev_last_activate      DATETIME,                                   -- last date/time event was activated
  site                  CHAR(32) NOT NULL,                          -- Site Name
  ev_guid               uniqueidentifier NOT NULL,                  -- Global unique record identifier
  ev_checksum           INTEGER NOT NULL,                           -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                           -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,                  -- Global unique record identifier
  ev_enable             SMALLINT NOT NULL,
  PRIMARY KEY           ( ev_partition, ev_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE externalip

```
(
  eip_id                INTEGER IDENTITY NOT FOR
                          REPLICATION UNIQUE,                       -- unique database ID
  eip_name              CHAR(32) NOT NULL,                          -- unique Pegasys name of the source
  eip_ip_address        CHAR(16),                                   -- IP address of the source (valid only if eip_computer_name is empty)
  eip_computer_name     CHAR(32),                                   -- DNS name of the source (valid only if eip_ip_address is empty)
  eip_allow             SMALLINT NOT NULL,                          -- 0 = Deny access, 1 = Allow access
  eip_routed            SMALLINT NOT NULL,                          -- 0 = not routed, 1 = Routed
  eip_usage_mask        INT NOT NULL,                               -- mask of usage
                                                                    -- 0x0001 BACNet
                                                                    -- 0x0002 XmlRpc
  site                  CHAR(32) NOT NULL,                          -- Site Name
  eip_guid              uniqueidentifier ROWGUIDCOL
                          UNIQUE NOT NULL,                          -- Global unique record identifier
  eip_checksum          INTEGER NOT NULL,                           -- Record checksum
  PRIMARY KEY           ( eip_id )
);
  CREATE UNIQUE INDEX   eip_site_name ON externalip (site, eip_name);
```

CREATE TABLE externalip_save

```
(
  eip_id                INTEGER NOT NULL,                           -- unique database ID
  eip_name              CHAR(32) NOT NULL,                          -- unique Pegasys name of the source
  eip_ip_address        CHAR(16),                                   -- IP address of the source (valid only if eip_computer_name is empty)
  eip_computer_name     CHAR(32),                                   -- DNS name of the source (valid only if eip_ip_address is empty)
  eip_allow             SMALLINT NOT NULL,                          -- 0 = Deny access, 1 = Allow access
  eip_routed            SMALLINT NOT NULL,                          -- 0 = not routed, 1 = Routed
  eip_usage_mask        INT,                                        -- mask of usage
                                                                    -- 0x0001 BACNet
```

```
                                                                    -- 0x0002 XmlRpc
  site                  CHAR(32) NOT NULL,                          -- Site Name
  eip_guid              uniqueidentifier NOT NULL,                  -- Global unique record identifier
  eip_checksum          INTEGER NOT NULL,                           -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                           -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,                  -- Global unique record identifier
  PRIMARY KEY           ( eip_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE external_trigger

```
(
  trigger_id            INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,    -- unique database ID
  trigger_text          CHAR(1024) NOT NULL,                            -- trigger text written by external system
);
```

CREATE TABLE facility

```
(
  f_name                        CHAR(32) NOT NULL,
  f_cardexpire_format           INTEGER NOT NULL,
  f_cardexpire_postype          INTEGER NOT NULL,
  f_cardexpire_position         INTEGER NOT NULL,
  f_cardexpire_thru             INTEGER NOT NULL,
  f_cardfacility_postype        INTEGER NOT NULL,
  f_cardFacility_position       INTEGER NOT NULL,
  f_cardid_postype              INTEGER NOT NULL,
  f_cardid_position             INTEGER NOT NULL,
  f_cardissue_postype           INTEGER NOT NULL,
  f_cardissue_position          INTEGER NOT NULL,
  f_cardtrack                   INTEGER NOT NULL,
  f_cardtrack_limit             INTEGER NOT NULL,
  f_dst_type                    INTEGER NOT NULL,
  f_dst_aheadmonth              INTEGER NOT NULL,
  f_dst_aheadsunday             INTEGER NOT NULL,
  f_dst_backmonth               INTEGER NOT NULL,
  f_dst_backsunday              INTEGER NOT NULL,
  f_facilitycode                CHAR(12) NULL,
  f_idcode_length               INTEGER NOT NULL,
  f_pin_length                  INTEGER NOT NULL,
  f_signonkey                   INTEGER NOT NULL,
  f_extended_access_flag        INTEGER NOT NULL,
  f_manager_flag                INTEGER NOT NULL,
  f_programmer_flag             INTEGER NOT NULL,
  f_extended_access_time        INTEGER NOT NULL,
  f_extended_shunt_time         INTEGER NOT NULL,
  f_create_keypad_id            INTEGER NOT NULL,
  f_database_name               CHAR(32) NOT NULL,
  f_card_format                 VARCHAR(4096) NULL,
  f_badge_type                  INTEGER NOT NULL,
  site                          CHAR(32) NOT NULL,                          -- Site Name
  f_guid                        uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,-- Global unique record identifier
  f_checksum                    INTEGER NOT NULL,                           -- Record checksum
  f_signon_badge_number         VARCHAR(100) NULL,
  f_signon_badge_issue          SMALLINT NULL,
  f_signon_badge_facility_code  INTEGER NULL,
  f_card_format_2               VARCHAR(4096) NULL,
  PRIMARY KEY                   (f_guid)
);
  CREATE UNIQUE INDEX           f_site_name ON facility (site, f_name);
```

CREATE TABLE facility_group

```
(
  fg_facility_guid      uniqueidentifier REFERENCES facility(f_guid)  NOT FOR REPLICATION,   -- facility ID
  fg_group_number       INTEGER,                                                             -- group number (1 - 32)
  fg_accgroup_id        INTEGER REFERENCES accgroup(ag_id)  NOT FOR REPLICATION,             -- access group ID
  fg_timezone_id        INTEGER REFERENCES timezone(tz_id)  NOT FOR REPLICATION,             -- timezone ID
  site                  CHAR(32) NOT NULL,                                                   -- Site Name
  fg_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                         -- Global unique record identifier
  fg_checksum           INTEGER NOT NULL,                                                    -- Record checksum
  PRIMARY KEY           ( fg_facility_guid, fg_group_number)
);
```

CREATE TABLE facility_group_save

```
(
  fg_facility_guid      uniqueidentifier,                                                    -- facility ID
  fg_group_number       INTEGER,                                                             -- group number (1 - 32)
  fg_accgroup_id        INTEGER,                                                             -- access group ID
  fg_timezone_id        INTEGER,                                                             -- timezone ID
  site                  CHAR(32) NOT NULL,                                                   -- Site Name
  fg_guid               uniqueidentifier NOT NULL,                                           -- Global unique record identifier
  fg_checksum           INTEGER NOT NULL,                                                    -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                                           -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                                   -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                                   -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                                    -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                                   -- Global unique record identifier
  PRIMARY KEY           ( fg_facility_guid, fg_group_number, upd_mode desc, upd_guid)
);
```

CREATE TABLE facility_hid

```
(
  fhid_id               INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,   -- Unique database ID of panel
  fhid_name             CHAR(32) NOT NULL,                             -- HID facility name
  fhid_ada_support_flag SMALLINT NOT NULL,                             -- 0 = none, 1 - 3 special flags A - C
  fhid_card_format      VARCHAR(4096) NULL,                            -- Card format
  site                  CHAR(32) NOT NULL,                             -- Site Name
  fhid_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
  fhid_checksum         INTEGER NOT NULL,                              -- Record checksum
  PRIMARY KEY           (fhid_id)
);
```

CREATE TABLE facility_hid_save

```
(
  fhid_id               INTEGER NOT NULL,                              -- Unique database ID of terminal
  fhid_name             CHAR(32) NOT NULL,                             -- HID facility name
  fhid_ada_support_flag SMALLINT NOT NULL,                             -- 0 = none, 1 - 3 special flags A - C
  fhid_card_format      VARCHAR(4096) NULL,                            -- Card format
  site                  CHAR(32) NOT NULL,                             -- Site Name
  fhid_guid             uniqueidentifier NOT NULL,                     -- Global unique record identifier
  fhid_checksum         INTEGER NOT NULL,                              -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                     -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                             -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                             -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                              -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                             -- Global unique record identifier
  PRIMARY KEY           (fhid_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE facility_save

```
(
  f_name                    CHAR(32) NOT NULL,
  f_cardexpire_format       INTEGER NOT NULL,
  f_cardexpire_postype      INTEGER NOT NULL,
  f_cardexpire_position     INTEGER NOT NULL,
  f_cardexpire_thru         INTEGER NOT NULL,
  f_cardfacility_postype    INTEGER NOT NULL,
  f_cardFacility_position   INTEGER NOT NULL,
  f_cardid_postype          INTEGER NOT NULL,
  f_cardid_position         INTEGER NOT NULL,
  f_cardissue_postype       INTEGER NOT NULL,
  f_cardissue_position      INTEGER NOT NULL,
  f_cardtrack               INTEGER NOT NULL,
  f_cardtrack_limit         INTEGER NOT NULL,
  f_dst_type                INTEGER NOT NULL,
  f_dst_aheadmonth          INTEGER NOT NULL,
  f_dst_aheadsunday         INTEGER NOT NULL,
  f_dst_backmonth           INTEGER NOT NULL,
  f_dst_backsunday          INTEGER NOT NULL,
  f_facilitycode            CHAR(12) NULL,
  f_idcode_length           INTEGER NOT NULL,
  f_pin_length              INTEGER NOT NULL,
  f_signonkey               INTEGER NOT NULL,
  f_manager_flag            INTEGER NOT NULL,
  f_extended_access_flag    INTEGER NOT NULL,
  f_programmer_flag         INTEGER NOT NULL,
  f_extended_access_time    INTEGER NOT NULL,
  f_extended_shunt_time     INTEGER NOT NULL,
  f_database_name           CHAR(32) NOT NULL,
  f_card_format             VARCHAR(4096) NULL,
  f_badge_type              INTEGER NOT NULL,
  f_create_keypad_id        INTEGER NOT NULL,
  site                      CHAR(32) NOT NULL,                    -- Site Name
  f_guid                    uniqueidentifier NOT NULL,            -- Global unique record identifier
  f_checksum                INTEGER NOT NULL,                     -- Record checksum
  upd_guid                  uniqueidentifier NOT NULL,            -- Guid for update operation
  upd_mode                  SMALLINT NOT NULL,                    -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp             DATETIME NOT NULL,                    -- Date / Time of the update
  upd_checksum              INTEGER NOT NULL,                     -- Checksum
  upd_rowguid               uniqueidentifier ROWGUIDCOL DEFAULT newid()
                              UNIQUE NOT NULL,                    -- Global unique record identifier
  f_signon_badge_number     VARCHAR(100) NULL,
  f_signon_badge_issue      SMALLINT NULL,
  f_signon_badge_facility_code INTEGER NULL,
  f_card_format_2           VARCHAR(4096) NULL,
  PRIMARY KEY               ( f_guid, upd_mode desc, upd_guid )
);
```

CREATE TABLE facility_status

```
(
  fs_guid                   uniqueidentifier ROWGUIDCOL REFERENCES facility(f_guid)
                              NOT FOR REPLICATION,                -- Global unique record identifier
  fs_last_xaction           INTEGER NULL,                         -- last transaction processed
  site                      CHAR(32) NOT NULL,                    -- Site Name
  fs_checksum               INTEGER NOT NULL,                     -- Record checksum
  PRIMARY KEY               (fs_guid)
);
```

CREATE TABLE fascn_ccc

```
(
  ccc_guid                  uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
```

```
  ccc_name                  CHAR(32) NOT NULL,                    -- CCC name
  ccc_card_id_rid           BINARY(5) NOT NULL,
  ccc_card_id_manufacturer  TINYINT NOT NULL,
  ccc_card_id_cardtype      TINYINT NOT NULL,
  ccc_card_id_cardid        VARCHAR(64) NOT NULL,
  ccc_container_version     TINYINT NOT NULL,
  ccc_grammer_version       TINYINT NOT NULL,
  ccc_card_url_rid          BINARY(5) NOT NULL,
  ccc_card_url_app_type     TINYINT NOT NULL,
  ccc_card_url_object_id    SMALLINT NOT NULL,
  ccc_card_url_app_id       SMALLINT NOT NULL,
  ccc_card_url_pin_id       TINYINT NOT NULL,
  ccc_pkcs15_version        TINYINT NOT NULL,
  ccc_data_model            TINYINT NOT NULL,
  ccc_card_apdus            BINARY(6) NOT NULL,
  site                      CHAR(32) NOT NULL,                    -- Site Name
  ccc_checksum              INTEGER NOT NULL,                     -- Record checksum
  PRIMARY KEY               ( ccc_guid )
);
```

CREATE TABLE fascn_ccc_save

```
(
  ccc_guid                  uniqueidentifier NOT NULL,            -- Global unique record identifier
  ccc_name                  CHAR(32) NOT NULL,                    -- CCC name
  ccc_card_id_rid           BINARY(5) NOT NULL,
  ccc_card_id_manufacturer  TINYINT NOT NULL,
  ccc_card_id_cardtype      TINYINT NOT NULL,
  ccc_card_id_cardid        VARCHAR(64) NOT NULL,
  ccc_container_version     TINYINT NOT NULL,
  ccc_grammer_version       TINYINT NOT NULL,
  ccc_card_url_rid          BINARY(5) NOT NULL,
  ccc_card_url_app_type     TINYINT NOT NULL,
  ccc_card_url_object_id    SMALLINT NOT NULL,
  ccc_card_url_app_id       SMALLINT NOT NULL,
  ccc_card_url_pin_id       TINYINT NOT NULL,
  ccc_pkcs15_version        TINYINT NOT NULL,
  ccc_data_model            TINYINT NOT NULL,
  ccc_card_apdus            BINARY(6) NOT NULL,
  site                      CHAR(32) NOT NULL,                    -- Site Name
  ccc_checksum              INTEGER NOT NULL,                     -- Record checksum
  upd_guid                  uniqueidentifier NOT NULL,            -- Guid for update operation
  upd_mode                  SMALLINT NOT NULL,                    -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp             DATETIME NOT NULL,                    -- Date / Time of the update
  upd_checksum              INTEGER NOT NULL,                     -- Checksum
  upd_rowguid               uniqueidentifier ROWGUIDCOL DEFAULT newid()
                              UNIQUE NOT NULL,                    -- Global unique record identifier
  PRIMARY KEY               ( ccc_guid, upd_guid )
);
```

CREATE TABLE fingerprint

```
(
  fp_cardholder_id          INTEGER,                              -- unique id of cardholder
  fp_create_date            DATETIME,                             -- date/time fingerprint was taken
  fp_image                  IMAGE,                                -- fingerprint image data
  fp_instance               INTEGER,                              -- fingerprint instance
  site                      CHAR(32) NOT NULL,                    -- Site Name
  fp_guid                   uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  fp_checksum               INTEGER NOT NULL,                     -- Record checksum
  PRIMARY KEY               ( fp_cardholder_id, fp_instance )
);
```

CREATE TABLE fingerprint_save

```
(
  fp_cardholder_id      INTEGER,                                          -- unique id of cardholder
  fp_create_date        DATETIME,                                         -- date/time fingerprint was taken
  fp_image              IMAGE,                                            -- fingerprint image data
  fp_instance           INTEGER,                                          -- fingerprint instance
  site                  CHAR(32) NOT NULL,                                -- Site Name
  fp_guid               uniqueidentifier NOT NULL,                        -- Global unique record identifier
  fp_checksum           INTEGER NOT NULL,                                 -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                        -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                 -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                -- Global unique record identifier
  PRIMARY KEY           ( fp_cardholder_id, fp_instance, upd_mode desc, upd_guid )
);
```

CREATE TABLE firealarm_entity

```
(
  fire_id int           IDENTITY NOT FOR REPLICATION                      unique,
  fire_type             int not NULL,
  fire_partition        INTEGER REFERENCES partition(part_number) NOT FOR REPLICATION,   -- ID of partition for this input
  fire_public           smallint,
  fire_opcid            varchar(32) NOT NULL,
  fire_name             varchar(32) NOT NULL,
  fire_opcname          varchar(255) unique NOT NULL,
  fire_parentid         int,
  site                  CHAR(32) NOT NULL,                                -- Site Name
  fire_querystring      varchar(64),
  fire_enable           smallint,
  fire_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,      -- Global unique record identifier
  fire_checksum         int NOT NULL,
  PRIMARY KEY           ( fire_id)
)
  CREATE UNIQUE INDEX   fire_opcname ON firealarm_entity( site, fire_opcname );
```

CREATE TABLE firealarm_entity_save

```
(
  fire_id               INTEGER NOT NULL,
  fire_type             int NOT NULL,
  fire_partition        INTEGER NOT NULL,
  fire_public           smallint,
  fire_opcid            varchar(32) NOT NULL,
  fire_name             varchar(32) NOT NULL,
  fire_opcname          varchar(255) unique NOT NULL,
  fire_parentid         int,
  site                  CHAR(32) NOT NULL,
  fire_querystring      varchar(64),
  fire_enable           smallint,
  fire_guid             uniqueidentifier NOT NULL,
  fire_checksum         int NOT NULL,
  upd_guid              uniqueidentifier NOT NULL,                        -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                 -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                -- Global unique record identifier
  PRIMARY KEY           (fire_id, upd_mode desc, upd_guid)
)
```

CREATE TABLE firealarm_status

```
(
  firestat_id           INTEGER REFERENCES firealarm_entity(fire_id)
                          NOT FOR REPLICATION,                          -- Fire Alarm ID to which this status belongs
  firestat_status       SMALLINT NOT NULL,                            -- Icon status
  firestat_substatus    SMALLINT NOT NULL,                            -- SubStatus
  firestat_timestamp    DATETIME NOT NULL,                            -- time of last current state (set by comms)
  site                  CHAR(32) NOT NULL,                            -- Site Name
  firestat_guid         uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  firestat_checksum     INTEGER NOT NULL,                             -- Record checksum
  PRIMARY KEY           (firestat_id)
);
```

CREATE TABLE flag_p900

```
(
  flp_id                INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,  -- Unique ID of this flag-- the flag  ID
  flp_name              varchar(32),                                  -- the flag name
  flp_number            SMALLINT NOT NULL,                            -- the flag number
  flp_status            SMALLINT NOT NULL,                            -- the flag status
  flp_report_change     SMALLINT NOT NULL,                            -- reports when changed
  flp_current_state     SMALLINT,                                     -- the current flag value
  flp_default_state     SMALLINT,                                     -- the default state
  flp_force_state       SMALLINT,                                     -- the force state
  flp_update_time       DATETIME,                                     -- The time stamp for control
  flp_query_string      CHAR(64),                                     -- query string enter by user
  flp_partition         INTEGER REFERENCES partition(part_number)
                          NOT FOR REPLICATION,                        -- ID of partition for this flag
  flp_public            SMALLINT NOT NULL,                            -- open to punlic
  flp_panel_id          INTEGER NOT NULL REFERENCES panel(p_panel_id)
                          NOT FOR REPLICATION,                        -- the panel ID
  site                  CHAR(32) NOT NULL,                            -- Site Name
  flp_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  flp_checksum          INTEGER NOT NULL,                             -- Record checksum
  PRIMARY KEY           (flp_partition, flp_id )
);
  CREATE INDEX          flp_public ON flag_p900(flp_public);
  CREATE UNIQUE INDEX   flp_number ON flag_p900 (flp_number, flp_panel_id);
  CREATE UNIQUE INDEX   flp_site_name ON flag_p900 (site, flp_name);
```

CREATE TABLE flag_p900_save

```
(
  flp_id                INTEGER NOT NULL,                             -- Unique ID of this flag -- the flag  ID
  flp_name              varchar(32),                                  -- the flag name
  flp_number            SMALLINT NOT NULL,                            -- the flag number
  flp_status            SMALLINT NOT NULL,                            -- the flag status
  flp_report_change     SMALLINT NOT NULL,                            -- reports when changed
  flp_current_state     SMALLINT,                                     -- the current flag value
  flp_default_state     SMALLINT,                                     -- the default state
  flp_force_state       SMALLINT,                                     -- the force state
  flp_update_time       DATETIME,                                     -- The time stamp for control
  flp_query_string      CHAR(64),                                     -- query string enter by user
  flp_partition         INTEGER NOT NULL,                             -- ID of partition for this flag
  flp_public            SMALLINT NOT NULL,                            -- open to punlic
  flp_panel_id          INTEGER NOT NULL,                             -- the panel ID
  site                  CHAR(32) NOT NULL,                            -- Site Name
  flp_guid              uniqueidentifier NOT NULL,                    -- Global unique record identifier
  flp_checksum          INTEGER NOT NULL,                             -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                    -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                            -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                            -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                             -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
```

```
  PRIMARY KEY          (flp_partition, flp_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE floorgrp

```
(
  fg_partition         INTEGER REFERENCES partition(part_number)
                         NOT FOR REPLICATION,                    -- partition ID to which this floor/door group belongs to
  fg_public            SMALLINT,                                 -- 1 = public, 0 = private
  fg_id                INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,  -- unique ID
  fg_name              CHAR(32) NOT NULL,                        -- group name
  fg_type              SMALLINT,                                 -- group type, 1 = elevator, 2 = cabinet
  site                 CHAR(32) NOT NULL,                        -- Site Name
  fg_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  fg_checksum          INTEGER NOT NULL,                         -- Record checksum
  PRIMARY KEY          (fg_partition, fg_id)
);
  CREATE UNIQUE INDEX      fg_name_type ON floorgrp (site, fg_name, fg_type);
  CREATE INDEX             fg_type ON floorgrp (fg_type);
```

CREATE TABLE floorgrpdata

```
(
  fgd_floorgrp_id      INTEGER REFERENCES floorgrp(fg_id) NOT FOR REPLICATION,          -- floor/door group id
  fgd_floormask_id     INTEGER REFERENCES floormask(fm_id) NOT FOR REPLICATION,         -- floor/door mask id
  fgd_elevator_id      INTEGER REFERENCES elevator(el_elevator_id) NOT FOR REPLICATION,  -- elevator/cabinet id
  site                 CHAR(32) NOT NULL,                                               -- Site Name
  fgd_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                     -- Global unique record identifier
  fgd_checksum         INTEGER NOT NULL,                                                -- Record checksum
  PRIMARY KEY          ( fgd_floorgrp_id, fgd_floormask_id, fgd_elevator_id)
);
```

CREATE TABLE floorgrpdata_save

```
(
  fgd_floorgrp_id      INTEGER,                                  -- floor/door group id
  fgd_floormask_id     INTEGER,                                  -- floor/door mask id
  fgd_elevator_id      INTEGER,                                  -- elevator/cabinet id
  site                 CHAR(32) NOT NULL,                        -- Site Name
  fgd_guid             uniqueidentifier NOT NULL,                -- Global unique record identifier
  fgd_checksum         INTEGER NOT NULL,                         -- Record checksum
  upd_guid             uniqueidentifier NOT NULL,                -- Guid for update operation
  upd_mode             SMALLINT NOT NULL,                        -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp        DATETIME NOT NULL,                        -- Date / Time of the update
  upd_checksum         INTEGER NOT NULL,                         -- Checksum
  upd_rowguid          uniqueidentifier ROWGUIDCOL DEFAULT newid()
                         UNIQUE NOT NULL,                        -- Global unique record identifier
  PRIMARY KEY          ( fgd_floorgrp_id, fgd_floormask_id, fgd_elevator_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE floorgrp_save

```
(
  fg_partition         INTEGER NOT NULL,                         -- partition ID to which this floor/door group belongs to
  fg_public            SMALLINT,                                 -- 1 = public, 0 = private
  fg_id                INTEGER NOT NULL,                         -- unique ID
  fg_name              CHAR(32) NOT NULL,                        -- group name
  fg_type              SMALLINT,                                 -- group type, 1 = elevator, 2 = cabinet
  site                 CHAR(32) NOT NULL,                        -- Site Name
  fg_guid              uniqueidentifier NOT NULL,                -- Global unique record identifier
  fg_checksum          INTEGER NOT NULL,                         -- Record checksum
  upd_guid             uniqueidentifier NOT NULL,                -- Guid for update operation
  upd_mode             SMALLINT NOT NULL,                        -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp        DATETIME NOT NULL,                        -- Date / Time of the update
```

```
  upd_checksum          INTEGER NOT NULL,                                   -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                  -- Global unique record identifier
  PRIMARY KEY           (fg_partition, fg_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE floormask

```
(
  fm_partition          INTEGER REFERENCES partition(part_number)
                          NOT FOR REPLICATION,          -- partition ID to which this floor/door mask belongs to
  fm_public             SMALLINT,                     -- 1 = public, 0 = private
  fm_id                 INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,  -- unique ID
  fm_name               CHAR(32) NOT NULL,            -- mask name
  fm_type               SMALLINT,                     -- mask type, 1 = elevator, 2 = cabinet
  site                  CHAR(32) NOT NULL,            -- Site Name
  fm_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  fm_checksum           INTEGER NOT NULL,             -- Record checksum
  PRIMARY KEY           (fm_partition, fm_id)
);
  CREATE UNIQUE INDEX   fm_name_type ON floormask (site, fm_name, fm_type);
  CREATE INDEX          fm_type ON floormask (fm_type);
```

CREATE TABLE floormaskdata

```
(
  fmd_mask_id           INTEGER REFERENCES floormask(fm_id) NOT FOR REPLICATION,   -- floor/door mask id
  fmd_floor_num         SMALLINT,
  site                  CHAR(32) NOT NULL,            -- Site Name
  fmd_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  fmd_checksum          INTEGER NOT NULL,             -- Record checksum
  PRIMARY KEY           (fmd_mask_id,fmd_floor_num)
);
```

CREATE TABLE floormaskdata_save

```
(
  fmd_mask_id           INTEGER NOT NULL,             -- floor/door mask id
  fmd_floor_num         SMALLINT,
  site                  CHAR(32) NOT NULL,            -- Site Name
  fmd_guid              uniqueidentifier NOT NULL,    -- Global unique record identifier
  fmd_checksum          INTEGER NOT NULL,             -- Record checksum

  upd_guid              uniqueidentifier NOT NULL,    -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,            -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,            -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,             -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,            -- Global unique record identifier
  PRIMARY KEY           (fmd_mask_id,fmd_floor_num, upd_mode desc, upd_guid)
);
```

CREATE TABLE floormask_save

```
(
  fm_partition          INTEGER NOT NULL,             -- partition ID to which this floor/door mask belongs to
  fm_public             SMALLINT,                     -- 1 = public, 0 = private
  fm_id                 INTEGER NOT NULL,             -- unique ID
  fm_name               CHAR(32) NOT NULL,            -- mask name
  fm_type               SMALLINT,                     -- mask type, 1 = elevator, 2 = cabinet
  site                  CHAR(32) NOT NULL,            -- Site Name
  fm_guid               uniqueidentifier NOT NULL,    -- Global unique record identifier
  fm_checksum           INTEGER NOT NULL,             -- Record checksum
```

```
  upd_guid              uniqueidentifier NOT NULL,                          -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                  -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                   -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                  -- Global unique record identifier
  PRIMARY KEY           (fm_partition, fm_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE floorname

```
(
  en_number             SMALLINT,                                           -- item number
  en_type               SMALLINT,                                           -- item type, 1 = elevator, 2 = cabinet
  en_name               CHAR(32) NULL,                                      -- item name
  site                  CHAR(32) NOT NULL,                                  -- Site Name
  en_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  en_checksum           INTEGER NOT NULL,                                   -- Record checksum
  PRIMARY KEY           (en_number, en_type, site)
);
  CREATE UNIQUE INDEX   en_name_type ON floorname (site, en_name, en_type);
```

CREATE TABLE floorname_save

```
(
  en_number             SMALLINT,                                           -- item number
  en_type               SMALLINT,                                           -- item type, 1 = elevator, 2 = cabinet
  en_name               CHAR(32) NULL,                                      -- item name
  site                  CHAR(32) NOT NULL,                                  -- Site Name
  en_guid               uniqueidentifier NOT NULL,                          -- Global unique record identifier
  en_checksum           INTEGER NOT NULL,                                   -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                          -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                  -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                   -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                  -- Global unique record identifier
  PRIMARY KEY           (en_number, en_type, site, upd_mode desc, upd_guid)
);
```

CREATE TABLE GenericText

```
(
  gt_id                 INTEGER IDENTITY NOT FOR REPLICATION PRIMARY KEY,   -- unique database ID
  gt_type               SMALLINT NOT NULL,                                  -- unique type number
  gt_number             SMALLINT NOT NULL,                                  -- unique format number
  gt_name               VARCHAR(64) NOT NULL,                               -- custom card format name
  gt_description        VARCHAR(128),                                       -- custom card format description
  gt_data               VARCHAR(7800) NOT NULL,                             -- card format image
  site                  CHAR(32) NOT NULL,                                  -- Site Name
  gt_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  gt_checksum           INTEGER NOT NULL,                                   -- Record checksum
);
  CREATE UNIQUE INDEX   gt_number ON GenericText (site, gt_type, gt_number);
```

CREATE TABLE GenericText_save

```
(
  gt_id                 INTEGER NOT NULL,                                   -- unique database ID
  gt_type               SMALLINT  NOT NULL,                                 -- unique type number
  gt_number             SMALLINT  NOT NULL,                                 -- unique format number
  gt_name               VARCHAR(64)  NOT NULL,                              -- custom card format name
  gt_description        VARCHAR(128),                                       -- custom card format description
  gt_data               VARCHAR(7800) NOT NULL,                             -- card format image
```

```
  site                CHAR(32) NOT NULL,                                  -- Site Name
  gt_guid             uniqueidentifier NOT NULL,                          -- Global unique record identifier
  gt_checksum         INTEGER NOT NULL,                                   -- Record checksum
  upd_guid            uniqueidentifier NOT NULL,                          -- Guid for update operation
  upd_mode            SMALLINT NOT NULL,                                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp       DATETIME NOT NULL,                                  -- Date / Time of the update
  upd_checksum        INTEGER NOT NULL,                                   -- Checksum
  upd_rowguid         uniqueidentifier ROWGUIDCOL DEFAULT newid()
                        UNIQUE NOT NULL,                                  -- Global unique record identifier
  PRIMARY KEY         (gt_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE HIDAgTzCombination

```
(
  hatc_id             INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,        -- unique database ID
  hatc_accgrp_id      INTEGER NOT NULL,                                   -- Access Group ID
  hatc_timezone_id    INTEGER NOT NULL,                                   -- Timezone ID
  site                CHAR(32) NOT NULL,                                  -- Site Name
  hatc_guid           uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  hatc_checksum       INTEGER NOT NULL,                                   -- Record checksum
  PRIMARY KEY         (hatc_id)
);

  CREATE INDEX        hatc_accgrp_id ON HIDAgTzCombination (hatc_accgrp_id);
  CREATE INDEX        hatc_timezone_id ON HIDAgTzCombination (hatc_timezone_id);
```

CREATE TABLE HIDAgTzPanelRef

```
(
  hatpr_panel_id      INTEGER NOT NULL,                                   -- Panel
  hatpr_reference     INTEGER NOT NULL,                                   -- Identifies AG/TZ combination no panel
  hatpr_accgrp_id     INTEGER NOT NULL,                                   -- Access Group ID
  hatpr_timezone_id   INTEGER NOT NULL,                                   -- Timezone ID
  site                CHAR(32) NOT NULL,                                  -- Site Name
  hatpr_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  hatpr_checksum      INTEGER NOT NULL,                                   -- Record checksum
  PRIMARY KEY         (hatpr_panel_id, hatpr_reference)
);
  CREATE INDEX        hatpr_accgrp_id ON HIDAgTzPanelRef (hatpr_accgrp_id);
  CREATE INDEX        hatpr_timezone_id ON HIDAgTzPanelRef (hatpr_timezone_id);
```

CREATE TABLE HIDAgTzPanelRefMap

```
(
  hatprm_panel_id     INTEGER NOT NULL,                                   -- Panel
  hatprm_reference    SMALLINT NOT NULL,                                  -- Identifies AG/TZ combination to panel
  hatprm_accgrp_id    INTEGER NOT NULL,                                   -- Access Group ID
  hatprm_accgrp_enabled SMALLINT NOT NULL,                                -- Access Group Enable
  hatprm_timezone_id  INTEGER NOT NULL,                                   -- Timezone ID
  site                CHAR(32) NOT NULL,                                  -- Site Name
  hatprm_guid         uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  hatprm_checksum     INTEGER NOT NULL,                                   -- Record checksum
  PRIMARY KEY         (hatprm_panel_id, hatprm_reference)
);
  CREATE INDEX        hatprm_accgrp_id ON HIDAgTzPanelRefMap (hatprm_accgrp_id);
  CREATE INDEX        hatprm_timezone_id ON HIDAgTzPanelRefMap (hatprm_timezone_id);
```

CREATE TABLE HIDAgTzReference

```
(
  hatr_panel_id       INTEGER NOT NULL,                                   -- Panel
  hatr_agtz_panel_ref SMALLINT NOT NULL,                                  -- Identifies AG/TZ combination to panel
  hatr_agtz_combo_id  INTEGER REFERENCES HIDAgTzCombination(hatc_id)
```

```
                        NOT FOR REPLICATION,                                    -- Identifies AG/TZ combo db entry
  site                  CHAR(32) NOT NULL,                                      -- Site Name
  hatr_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,            -- Global unique record identifier
  hatr_checksum         INTEGER NOT NULL,                                       -- Record checksum
  PRIMARY KEY (hatr_panel_id)
);
  CREATE INDEX          hatr_panel_id ON HIDAgTzReference (hatr_panel_id);
-- hatr_agtz_panel_ref used by query when finding next free reference
  CREATE INDEX          hatr_agtz_panel_ref ON HIDAgTzReference (hatr_agtz_panel_ref);
```

CREATE TABLE holiday

```
(
  ho_partition          INTEGER REFERENCES partition(part_number)
                          NOT FOR REPLICATION,                                  -- partition ID to which this holiday belongs
  ho_public             SMALLINT,                                               -- 1 = public, 0 = private
  ho_id                 INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,            -- unique ID for this holiday
  ho_name               CHAR(32) NOT NULL,                                      -- holiday name
  ho_date               DATETIME,                                               -- date of holiday
  ho_type               SMALLINT,                                               -- type 0, 1, or 2
  site                  CHAR(32) NOT NULL,                                      -- Site Name
  ho_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,            -- Global unique record identifier
  ho_checksum           INTEGER NOT NULL,                                       -- Record checksum
  PRIMARY KEY           ( ho_partition, ho_id )
);
  CREATE UNIQUE INDEX   ho_part_name ON holiday (site, ho_partition, ho_name );
  CREATE INDEX          ho_public ON holiday ( ho_public );
  CREATE UNIQUE INDEX   ho_site_name ON holiday (site, ho_name);
```

CREATE TABLE holiday_save

```
(
  ho_partition          INTEGER NOT NULL,                                       -- partition ID to which this holiday belongs
  ho_public             SMALLINT,                                               -- 1 = public, 0 = private
  ho_id                 INTEGER NOT NULL,                                       -- unique ID for this holiday
  ho_name               CHAR(32) NOT NULL,                                      -- holiday name
  ho_date               DATETIME,                                               -- date of holiday
  ho_type               SMALLINT,                                               -- type 0, 1, or 2
  site                  CHAR(32) NOT NULL,                                      -- Site Name
  ho_guid               uniqueidentifier NOT NULL,                              -- Global unique record identifier
  ho_checksum           INTEGER NOT NULL,                                       -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                              -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                      -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                      -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                       -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                      -- Global unique record identifier
  PRIMARY KEY           ( ho_partition, ho_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE hoursonsiteterminals

```
(
  host_zone_id          INTEGER REFERENCES hoursonsitezones(hosz_zone_id)
                          NOT FOR REPLICATION,                                  -- unique database ID of zone
  host_term_id          INTEGER REFERENCES terminal(tp_term_id) NOT FOR REPLICATION,  -- terminal ID
  host_entryexit        SMALLINT NOT NULL,                                      -- 1 = entry, 0 = exit
  site                  CHAR(32) NOT NULL,                                      -- Site Name
  host_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,            -- Global unique record identifier
  host_checksum         INTEGER NOT NULL,                                       -- Record checksum
  PRIMARY KEY           (host_zone_id, host_term_id)
);
```

CREATE TABLE hoursonsiteterminals_save

```
(
  host_zone_id          INTEGER,                                          -- unique database ID
  host_term_id          INTEGER,                                          -- terminal ID
  host_entryexit        SMALLINT NOT NULL,                                -- 1 = entry, 0 = exit
  site                  CHAR(32) NOT NULL,                                -- Site Name
  host_guid             uniqueidentifier NOT NULL,                        -- Global unique record identifier
  host_checksum         INTEGER NOT NULL,                                 -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                        -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                 -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                -- Global unique record identifier
  PRIMARY KEY           ( host_zone_id, host_term_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE hoursonsitezones

```
(
  hosz_partition            INTEGER REFERENCES partition(part_number)
                              NOT FOR REPLICATION,                      -- partition ID to which this area belongs
  hosz_public               SMALLINT,                                   -- 1 = area is public, 0 = area is private
  hosz_zone_id              INTEGER IDENTITY NOT FOR REPLICATION UNIQUE, -- unique database ID
  hosz_zone_name            CHAR(32) NOT NULL,                          -- Zone Name
  site                      CHAR(32) NOT NULL,                          -- Site Name
  hosz_guid                 uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  hosz_checksum             INTEGER NOT NULL,                           -- Record checksum
  PRIMARY KEY               (hosz_partition, hosz_zone_id)
);
  CREATE UNIQUE INDEX       hosz_site_name ON hoursonsitezones (site, hosz_zone_name);
```

CREATE TABLE hoursonsitezones_save

```
(
  hosz_partition        INTEGER,                                          -- partition ID to which this area belongs
  hosz_public           SMALLINT,                                         -- 1 = area is public, 0 = area is private
  hosz_zone_id          INTEGER,                                          -- unique database ID
  hosz_zone_name        CHAR(32) NOT NULL,                                -- Zone Name
  site                  CHAR(32) NOT NULL,                                -- Site Name
  hosz_guid             uniqueidentifier NOT NULL,                        -- Global unique record identifier
  hosz_checksum         INTEGER NOT NULL,                                 -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                        -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                 -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                -- Global unique record identifier
  PRIMARY KEY           ( hosz_partition, hosz_zone_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE icon_image

```
(
  image_id              INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,      -- unique ID for this icon image
  image_set_id          INTEGER REFERENCES icon_image_set(set_id)
                          NOT FOR REPLICATION,                            -- image set ID that this image is for
  image_state           SMALLINT,                                         -- state this icon displays (see paneldb.h, terminaldb.h,
                                                                            inputptdb.h, outputptdb.h)
  image_data            IMAGE,                                            -- icon image data
  site                  CHAR(32) NOT NULL,                                -- Site Name
  image_guid            uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,      -- Global unique record identifier
  image_checksum        INTEGER NOT NULL,                                 -- Record checksum
  PRIMARY KEY           ( image_id )
```

```
);
  CREATE INDEX        image_set_id ON icon_image ( image_set_id );
```

CREATE TABLE icon_image_save

```
(
  image_id            INTEGER NOT NULL,                          -- unique ID for this icon image
  image_set_id        INTEGER NOT NULL,                          -- image set ID that this image is for
  image_state         SMALLINT,                                  -- state this icon displays (see paneldb.h, terminaldb.h,
                                                                    inputptdb.h, outputptdb.h)
  image_data          IMAGE,                                     -- icon image data
  site                CHAR(32) NOT NULL,                         -- Site Name
  image_guid          uniqueidentifier NOT NULL,                 -- Global unique record identifier
  image_checksum      INTEGER NOT NULL,                          -- Record checksum
  upd_guid            uniqueidentifier NOT NULL,                 -- Guid for update operation
  upd_mode            SMALLINT NOT NULL,                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp       DATETIME NOT NULL,                         -- Date / Time of the update
  upd_checksum        INTEGER NOT NULL,                          -- Checksum
  upd_rowguid         uniqueidentifier ROWGUIDCOL DEFAULT newid()
                        UNIQUE NOT NULL,                         -- Global unique record identifier
  PRIMARY KEY         ( image_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE icon_image_set

```
(
  set_partition       INTEGER REFERENCES partition(part_number)
                        NOT FOR REPLICATION,                     -- partition ID to which this set belongs
  set_public          SMALLINT,                                  -- 1 = public, 0 = private
  set_id              INTEGER IDENTITY NOT FOR REPLICATION UNIQUE, -- unique ID for this image set
  set_name            CHAR(64) NOT NULL,                         -- image set name
  set_type            SMALLINT,                                  -- image set type (see iconimagesetdb.h)
  site                CHAR(32) NOT NULL,                         -- Site Name
  set_guid            uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  set_checksum        INTEGER NOT NULL,                          -- Record checksum
  set_default_set     SMALLINT NOT NULL,                         -- 1 = default image set
  PRIMARY KEY         ( set_partition, set_id )
);
  CREATE INDEX        set_public ON icon_image_set ( set_public );
```

CREATE TABLE icon_image_set_save

```
(
  set_partition       INTEGER NOT NULL,                          -- partition ID to which this set belongs
  set_public          SMALLINT,                                  -- 1 = public, 0 = private
  set_id              INTEGER NOT NULL,                          -- unique ID for this image set
  set_name            CHAR(64) NOT NULL,                         -- image set name
  set_type            SMALLINT,                                  -- image set type (see iconimagesetdb.h)
  site                CHAR(32) NOT NULL,                         -- Site Name
  set_guid            uniqueidentifier NOT NULL,                 -- Global unique record identifier
  set_checksum        INTEGER NOT NULL,                          -- Record checksum
  upd_guid            uniqueidentifier NOT NULL,                 -- Guid for update operation
  upd_mode            SMALLINT NOT NULL,                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp       DATETIME NOT NULL,                         -- Date / Time of the update
  upd_checksum        INTEGER NOT NULL,                          -- Checksum
  upd_rowguid         uniqueidentifier ROWGUIDCOL DEFAULT newid()
                        UNIQUE NOT NULL,                         -- Global unique record identifier
  set_default_set     SMALLINT NOT NULL,                         -- 1 = default image set
  PRIMARY KEY         ( set_partition, set_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE icon_overlay

```
(
   overlay_id           INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,        -- partition ID to which this icon belongs
   icon_x_pos           SMALLINT,                                           -- map x position of this icon
   icon_y_pos           SMALLINT,                                           -- map y position of this icon
   item_id              INTEGER,                                            -- item ID that this icon represents
   icon_map_id          INTEGER REFERENCES map(map_id) NOT FOR REPLICATION,  -- map ID that contains this icon
   set_id               INTEGER REFERENCES icon_image_set(set_id)
                           NOT FOR REPLICATION,                             -- icon image set ID for this icon
   set_type             SMALLINT,                                           -- icon image set type 0-5
   label                CHAR(255),
   event_1              INTEGER NULL REFERENCES event(ev_id) NOT FOR REPLICATION,
   event_2              INTEGER NULL REFERENCES event(ev_id) NOT FOR REPLICATION,
   event_3              INTEGER NULL REFERENCES event(ev_id) NOT FOR REPLICATION,
   event_4              INTEGER NULL REFERENCES event(ev_id) NOT FOR REPLICATION,
   event_label_1        CHAR(32) NULL,
   event_label_2        CHAR(32) NULL,
   event_label_3        CHAR(32) NULL,
   event_label_4        CHAR(32) NULL,
   site                 CHAR(32) NOT NULL,                                  -- Site Name
   guid                 uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
   checksum             INTEGER NOT NULL,                                   -- Record checksum
   text_color           INTEGER NOT NULL,
   text_bkgnd_color     INTEGER NOT NULL,
   text_position        INTEGER NOT NULL,
   text_font            CHAR(64) NULL,
   text_size            SMALLINT NULL,
   text_weight          SMALLINT NULL,
   text_italic          SMALLINT NULL,
   PRIMARY KEY          ( overlay_id )
);
   CREATE INDEX         item_id ON icon_overlay ( item_id );
   CREATE INDEX         icon_map_id ON icon_overlay ( icon_map_id );
   CREATE INDEX         set_type ON icon_overlay ( set_type );
```

CREATE TABLE icon_overlay_save

```
(
   overlay_id           INTEGER NOT NULL,                                   -- partition ID to which this icon belongs
   icon_x_pos           SMALLINT,                                           -- map x position of this icon
   icon_y_pos           SMALLINT,                                           -- map y position of this icon
   item_id              INTEGER,                                            -- item ID that this icon represents
   icon_map_id          INTEGER NULL,                                       -- map ID that contains this icon
   set_id               INTEGER NULL,                                       -- icon image set ID for this icon
   set_type             SMALLINT,                                           -- icon image set type 0-5
   label                CHAR(255),
   event_1              INTEGER NULL,
   event_2              INTEGER NULL,
   event_3              INTEGER NULL,
   event_4              INTEGER NULL,
   event_label_1        CHAR(32) NULL,
   event_label_2        CHAR(32) NULL,
   event_label_3        CHAR(32) NULL,
   event_label_4        CHAR(32) NULL,
   site                 CHAR(32) NOT NULL,                                  -- Site Name
   guid                 uniqueidentifier NOT NULL,                          -- Global unique record identifier
   checksum             INTEGER NOT NULL,                                   -- Record checksum
   upd_guid             uniqueidentifier NOT NULL,                          -- Guid for update operation
   upd_mode             SMALLINT NOT NULL,                                  -- Update mode (0 - delete, 1 - edit, 2 - new)
   upd_timestamp        DATETIME NOT NULL,                                  -- Date / Time of the update
   upd_checksum         INTEGER NOT NULL,                                   -- Checksum
   upd_rowguid          uniqueidentifier ROWGUIDCOL DEFAULT newid()
                           UNIQUE NOT NULL,                                 -- Global unique record identifier
   text_color           INTEGER NULL,
```

```
    text_bkgnd_color        INTEGER NULL,
    text_position           INTEGER NULL,
    text_font               CHAR(64) NULL,
    text_size               SMALLINT NULL,
    text_weight             SMALLINT NULL,
    text_italic             SMALLINT NULL,
    PRIMARY KEY             ( overlay_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE idbadge

```
(
    ib_partition            INTEGER REFERENCES partition(part_number)
                                NOT FOR REPLICATION,        -- partition ID to which this badge belongs
    ib_public               SMALLINT,                       -- 1 = public, 0 = private
    ib_number_str           VARCHAR(100) NOT NULL,          -- badge number
    ib_cardholder_id        INTEGER REFERENCES cardholder(c_id) NOT FOR REPLICATION,  -- cardholder ID that has this badge
    ib_alpha                CHAR(4) NULL,                   -- alpha field of badge
    ib_issue                SMALLINT,                       -- issue level of badge
    ib_description          CHAR(64) NULL,                  -- user entered description
    ib_disabled             SMALLINT,                       -- 1 = disabled, 0 = active
    ib_start_timestamp      DATETIME,                       -- start timestamp
    ib_exp_timestamp        DATETIME,                       -- expiration timestamp
    ib_reason               CHAR(32) NULL,                  -- reason for badge issue
    ib_design               INTEGER NULL REFERENCES badgelayout(bl_id)
                                NOT FOR REPLICATION,        -- badge design ID
    site                    CHAR(32) NOT NULL,              -- Site Name
    ib_guid                 uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
    ib_checksum             INTEGER NOT NULL,               -- Record checksum
    ib_purpose              uniqueidentifier REFERENCES badgepurpose(bp_guid)
                                NOT FOR REPLICATION,
    ib_data_style           SMALLINT NOT NULL,              -- 0 = normal, 1 = FASCN
    ib_operator             CHAR(32),                       -- Operator that last modified record
    ib_edit_time            DATETIME,                       -- Time this record was last edited
    PRIMARY KEY             ( ib_number_str, ib_cardholder_id )
);
    CREATE INDEX            ib_partition ON idbadge ( ib_partition );
    CREATE INDEX            ib_public ON idbadge ( ib_public );
```

CREATE TABLE idbadge_save

```
(
    ib_partition            INTEGER NOT NULL,               -- partition ID to which this badge belongs
    ib_public               SMALLINT,                       -- 1 = public, 0 = private
    ib_number_str           VARCHAR(100) NOT NULL,          -- badge number
    ib_cardholder_id        INTEGER NOT NULL,               -- cardholder ID that has this badge
    ib_alpha                CHAR(4) NULL,                   -- alpha field of badge
    ib_issue                SMALLINT,                       -- issue level of badge
    ib_description          CHAR(64) NULL,                  -- user entered description
    ib_disabled             SMALLINT,                       -- 1 = disabled, 0 = active
    ib_start_timestamp      DATETIME,                       -- start timestamp
    ib_exp_timestamp        DATETIME,                       -- expiration timestamp
    ib_reason               CHAR(32) NULL,                  -- reason for badge issue
    ib_design               INTEGER NULL,                   -- badge design ID
    site                    CHAR(32) NOT NULL,              -- Site Name
    ib_guid                 uniqueidentifier NOT NULL,      -- Global unique record identifier
    ib_checksum             INTEGER NOT NULL,               -- Record checksum
    upd_guid                uniqueidentifier NOT NULL,      -- Guid for update operation
    upd_mode                SMALLINT NOT NULL,              -- Update mode (0 - delete, 1 - edit, 2 - new)
    upd_timestamp           DATETIME NOT NULL,              -- Date / Time of the update
    upd_checksum            INTEGER NOT NULL,               -- Checksum
    upd_rowguid             uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,  -- Global unique record identifier
    ib_purpose              uniqueidentifier NULL,
    ib_data_style           SMALLINT NOT NULL,              -- 0 = normal, 1 = FASCN
```

```
  ib_operator           CHAR(32),                                          -- Operator that last modified record
  ib_edit_time          DATETIME,                                          -- Time this record was last edited
  PRIMARY KEY           ( ib_number_str, ib_cardholder_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE [dbo].[IDSequences]

```
(
  [SeqName]             [char] (10) NULL,
  [ID]                  [int] NULL,
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  guid                  uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  checksum              INTEGER NOT NULL,                                  -- Record checksum
  ) ON [PRIMARY]
;
```

CREATE TABLE [IDSequences_save]

```
(
  [SeqName]             [char] (10) NULL,
  [ID]                  [int] NULL,
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  guid                  uniqueidentifier NOT NULL,                         -- Global unique record identifier
  checksum              INTEGER NOT NULL,                                  -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                         -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                 -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                  -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                 -- Global unique record identifier
  PRIMARY KEY           (guid, upd_mode desc, upd_guid)
) ON [PRIMARY]
;
```

CREATE TABLE igdata

```
(
  ig_group_id           INTEGER REFERENCES inputgrp(ip_group_id) NOT FOR REPLICATION,   -- input group ID
  ig_inputpt_id         INTEGER REFERENCES input(ip_point_id) NOT FOR REPLICATION,      -- input point ID
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  ig_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  ig_checksum           INTEGER NOT NULL,                                  -- Record checksum
  PRIMARY KEY           ( ig_group_id, ig_inputpt_id )
);
```

CREATE TABLE igdata_save

```
(
  ig_group_id           INTEGER,                                           -- input group ID
  ig_inputpt_id         INTEGER,                                           -- input point ID
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  ig_guid               uniqueidentifier NOT NULL,                         -- Global unique record identifier
  ig_checksum           INTEGER NOT NULL,                                  -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                         -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                 -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                  -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,   -- Global unique record identifier
  PRIMARY KEY           ( ig_group_id, ig_inputpt_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE imagerecallfilter

```
(
  partition          INTEGER REFERENCES partition(part_number)
                       NOT FOR REPLICATION,                                  -- partition ID to which this filter belongs
  public_flag        SMALLINT,                                               -- 1 = public, 0 = private
  id                 INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,            -- unique ID for this filter
  name               CHAR(32) NOT NULL,                                      -- filter name
  access_grant       SMALLINT,                                               -- 1 = display upon access grant
  any_deny           SMALLINT,                                               -- 1 = display upon access deny
  invalid_card       SMALLINT,                                               -- 1 = display upon invalid card
  invalid_issue      SMALLINT,                                               -- 1 = display upon invalid issue
  invalid_pin        SMALLINT,                                               -- 1 = display upon invalid pin
  antipassback       SMALLINT,                                               -- 1 = display upon antipassback
  card_tz            SMALLINT,                                               -- 1 = display upon invalid card timezone
  invalid_reader     SMALLINT,                                               -- 1 = display upon invalid reader
  all_terminals      SMALLINT,                                               -- 1 = display for any terminal
  all_term_grps      SMALLINT,                                               -- 1 = display for any terminal group
  terminal           INTEGER REFERENCES terminal(tp_term_id) NOT FOR REPLICATION,  -- terminal ID to display for
  term_group         INTEGER REFERENCES termgrp(tg_id) NOT FOR REPLICATION,  -- terminal group ID to display for
  site               CHAR(32) NOT NULL,                                      -- Site Name
  guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,            -- Global unique record identifier
  checksum           INTEGER NOT NULL,                                       -- Record checksum
  access_duress      SMALLINT,                                               -- 1 = display upon access duress grant
  PRIMARY KEY        ( partition, id )
);
  CREATE INDEX       name ON imagerecallfilter ( name );
```

CREATE TABLE imagerecallfilter_save

```
(
  partition          INTEGER NOT NULL,                                       -- partition ID to which this filter belongs
  public_flag        SMALLINT,                                               -- 1 = public, 0 = private
  id                 INTEGER NOT NULL,                                       -- unique ID for this filter
  name               CHAR(32) NOT NULL,                                      -- filter name
  access_grant       SMALLINT,                                               -- 1 = display upon access grant
  any_deny           SMALLINT,                                               -- 1 = display upon access deny
  invalid_card       SMALLINT,                                               -- 1 = display upon invalid card
  invalid_issue      SMALLINT,                                               -- 1 = display upon invalid issue
  invalid_pin        SMALLINT,                                               -- 1 = display upon invalid pin
  antipassback       SMALLINT,                                               -- 1 = display upon antipassback
  card_tz            SMALLINT,                                               -- 1 = display upon invalid card timezone
  invalid_reader     SMALLINT,                                               -- 1 = display upon invalid reader
  all_terminals      SMALLINT,                                               -- 1 = display for any terminal
  all_term_grps      SMALLINT,                                               -- 1 = display for any terminal group
  terminal           INTEGER,                                                -- terminal ID to display for
  term_group         INTEGER,                                                -- terminal group ID to display for
  site               CHAR(32) NOT NULL,                                      -- Site Name
  guid               uniqueidentifier NOT NULL,                              -- Global unique record identifier
  checksum           INTEGER NOT NULL,                                       -- Record checksum
  upd_guid           uniqueidentifier NOT NULL,                              -- Guid for update operation
  upd_mode           SMALLINT NOT NULL,                                      -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp      DATETIME NOT NULL,                                      -- Date / Time of the update
  upd_checksum       INTEGER NOT NULL,                                       -- Checksum
  upd_rowguid        uniqueidentifier ROWGUIDCOL DEFAULT newid()
                       UNIQUE NOT NULL,                                      -- Global unique record identifier
  access_duress      SMALLINT,                                               -- 1 = display upon access duress grant
  PRIMARY KEY        ( partition, id, upd_mode desc, upd_guid )
);
```

CREATE TABLE input

```
(
  ip_partition       INTEGER REFERENCES partition(part_number)
                       NOT FOR REPLICATION,                                  -- ID of partition for this input
```

```
    ip_public           SMALLINT,                                       -- publication
    ip_point_id         INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,    -- Unique database ID of input point
    ip_point_name       varchar(32),                                    -- name of input point
    ip_point_number     SMALLINT,                                       -- input number
    ip_term_id          INTEGER NULL REFERENCES terminal(tp_term_id)
                          NOT FOR REPLICATION,                          -- terminal ID that has this input, is null if this is a
                                                                        -- panel soft input, if null then ip_panel_id is used
    ip_panel_id         INTEGER NULL REFERENCES panel(p_panel_id)
                          NOT FOR REPLICATION,                          -- panel ID of this input, is used only for panel soft
                                                                          inputs -- this is null if ip_term_id is used -- only
                                                                          1 of these fields can be used for any given input point
    ip_state            SMALLINT,                                       -- input state
    ip_enable_flag      SMALLINT,                                       -- input enable
    ip_query_string     CHAR(64),                                       -- query string enter by user
    ip_soft_input       SMALLINT,                                       -- 1 = soft input, 0 = hardware input
    ip_suppress_tz      INTEGER NULL REFERENCES timezone(tz_id)
                          NOT FOR REPLICATION,                          -- timezone ID when input suppressed (always active
                                                                          if NULL)
    site                CHAR(32) NOT NULL,                              -- Site Name
    ip_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
    ip_checksum         INTEGER NOT NULL,                               -- Record checksum
    PRIMARY KEY         ( ip_partition, ip_point_id )
);
    CREATE INDEX            ip_public ON input(ip_public);
    CREATE INDEX            ip_term_id ON input ( ip_term_id );
    CREATE UNIQUE INDEX     ip_term_panel_num ON input ( ip_term_id, ip_panel_id, ip_point_number );
    CREATE UNIQUE INDEX     ip_site_point_name ON input ( site,ip_point_name );
```

CREATE TABLE Input2CamInfo

```
(
    i2c_id              INTEGER IDENTITY NOT FOR REPLICATION,
    i2c_input_point_id  INTEGER NULL REFERENCES input(ip_point_id),
    i2c_Partition       INTEGER NOT NULL REFERENCES partition(part_number),   -- Partition ID to which this mapping belongs
    i2c_Public          INTEGER NOT NULL,                                     -- 1 = public, 0 = private
    i2c_terminal_id     INTEGER NULL REFERENCES terminal(tp_term_id),
    i2c_description     VARCHAR(256) NOT NULL,
    i2c_channel_id      INTEGER NOT NULL REFERENCES AVChannel(CCH_ID),
    i2c_preset_id       INTEGER NULL REFERENCES AVPreset(CAP_ID),
    site                CHAR(32) NOT NULL,                                    -- Site Name
    i2c_guid            uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                    -- Global unique record identifier
    i2c_checksum        INTEGER NOT NULL,                                     -- Record checksum
    PRIMARY KEY         ( i2c_id )
)
    CREATE INDEX        i2c_input_point_id ON Input2CamInfo(i2c_input_point_id);
    CREATE INDEX        i2c_terminal_id ON Input2CamInfo(i2c_terminal_id);
```

CREATE TABLE dbo.Input2CamInfo_save

```
(
    i2c_id int NOT NULL,
    i2c_Partition       INTEGER NOT NULL,                                     -- Partition ID to which this mapping belongs
    i2c_Public          INTEGER NOT NULL,                                     -- 1 = public, 0 = private
    i2c_input_point_id  int NULL,
    i2c_terminal_id     int NULL,
    i2c_description     VARCHAR(256) NULL,
    i2c_channel_id      int NULL,
    i2c_preset_id       int NULL,
    site                char (32) NOT NULL,
    i2c_guid            uniqueidentifier NOT NULL,
    i2c_checksum        int NOT NULL,
    upd_guid            uniqueidentifier NOT NULL,                            -- Guid for update operation
    upd_mode            SMALLINT NOT NULL,                                    -- Update mode (0 - delete, 1 - edit, 2 - new)
```

```
  upd_timestamp          DATETIME NOT NULL,                                  -- Date / Time of the update
  upd_checksum           INTEGER NOT NULL,                                   -- Checksum
  upd_rowguid            uniqueidentifier ROWGUIDCOL DEFAULT newid()
                           UNIQUE NOT NULL,                                  -- Global unique record identifier
  PRIMARY KEY            (i2c_id, upd_mode desc, upd_guid)
)
```

CREATE TABLE inputgrp

```
(
  ip_group_partition     INTEGER REFERENCES partition(part_number)
                           NOT FOR REPLICATION,                              -- partition ID to which this group belongs
  ip_group_public        SMALLINT,                                           -- 1 = public, 0 = private
  ip_group_id            INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,        -- unique ID for this group
  ip_group_name          CHAR(32) NOT NULL,                                  -- group name
  ip_group_panel_id      INTEGER REFERENCES panel(p_panel_id) NOT FOR REPLICATION,  -- panel ID to which this group belongs
  ip_group_index         SMALLINT,                                           -- panel index of group (1-598)
  site                   CHAR(32) NOT NULL,                                  -- Site Name
  ip_group_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  ip_group_checksum      INTEGER NOT NULL,                                   -- Record checksum
  PRIMARY KEY            ( ip_group_partition, ip_group_id)
);
   CREATE INDEX              ip_group_public ON inputgrp ( ip_group_public );
  CREATE INDEX               ip_group_panel_id ON inputgrp ( ip_group_panel_id);
  CREATE INDEX               ip_group_index ON inputgrp ( ip_group_index);
  CREATE UNIQUE INDEX        ip_site_name ON inputgrp (site, ip_group_name);
```

CREATE TABLE inputgrp_save

```
(
  ip_group_partition     INTEGER NOT NULL,                                   -- partition ID to which this group belongs
  ip_group_public        SMALLINT,                                           -- 1 = public, 0 = private
  ip_group_id            INTEGER NOT NULL,                                   -- unique ID for this group
  ip_group_name          CHAR(32) NOT NULL,                                  -- group name
  ip_group_panel_id      INTEGER,                                            -- panel ID to which this group belongs
  ip_group_index         SMALLINT,                                           -- panel index of group (1-598)
  site                   CHAR(32) NOT NULL,                                  -- Site Name
  ip_group_guid          uniqueidentifier NOT NULL,                          -- Global unique record identifier
  ip_group_checksum      INTEGER NOT NULL,                                   -- Record checksum
  upd_guid               uniqueidentifier NOT NULL,                          -- Guid for update operation
  upd_mode               SMALLINT NOT NULL,                                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp          DATETIME NOT NULL,                                  -- Date / Time of the update
  upd_checksum           INTEGER NOT NULL,                                   -- Checksum
  upd_rowguid            uniqueidentifier ROWGUIDCOL DEFAULT newid()
                           UNIQUE NOT NULL,                                  -- Global unique record identifier
  PRIMARY KEY            ( ip_group_partition, ip_group_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE input_status

```
(
  ipstat_point_id        INTEGER REFERENCES input(ip_point_id) NOT FOR REPLICATION,  -- input ID to which this status belongs
  ipstat_status          SMALLINT NOT NULL,                                  -- current state for input (set by comms)
  ipstat_timestamp       DATETIME NOT NULL,                                  -- time of last current state (set by comms)
  site                   CHAR(32) NOT NULL,                                  -- Site Name
  ipstat_guid            uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  ipstat_checksum        INTEGER NOT NULL,                                   -- Record checksum
  PRIMARY KEY            (ipstat_point_id)
);
```

CREATE TABLE input_ck

```
(
  ip_id                  INTEGER REFERENCES input(ip_point_id) NOT FOR REPLICATION,  -- Unique database ID of input
```

```
  ip_delay_enable       SMALLINT,                                          -- 1 = alarm delay enabled
  ip_relay_flag         SMALLINT,                                          -- 1 = set panel relay when active
  ip_link_type          SMALLINT,                                          -- I/O link type (0-7)
  ip_in_out_link_grp    INTEGER NULL REFERENCES outputgrp(op_group_id)
                          NOT FOR REPLICATION,                             -- I/O link output group ID
  ip_exit_delay         SMALLINT,                                          -- entry/exit delay in secs (0-600)
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  ip_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  ip_checksum           INTEGER NOT NULL,                                  -- Record checksum
  PRIMARY KEY           (ip_id)
);
```

CREATE TABLE input_ck_save

```
(
  ip_id                 INTEGER NOT NULL,                                  -- Unique database ID of input
  ip_delay_enable       SMALLINT,                                          -- 1 = alarm delay enabled
  ip_relay_flag         SMALLINT,                                          -- 1 = set panel relay when active
  ip_link_type          SMALLINT,                                          -- I/O link type (0-7)
  ip_in_out_link_grp    INTEGER NULL,                                      -- I/O link output group ID
  ip_exit_delay         SMALLINT,                                          -- entry/exit delay in secs (0-600)
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  ip_guid               uniqueidentifier NOT NULL,                         -- Global unique record identifier
  ip_checksum           INTEGER NOT NULL,                                  -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                         -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                 -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                  -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                 -- Global unique record identifier
  PRIMARY KEY           (ip_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE input_hid

```
(
  ipphid_input_id               INTEGER REFERENCES input(ip_point_id)
                                  NOT FOR REPLICATION,                     -- ID of input
  ipphid_debounce_millisec      SMALLINT NOT NULL,                         -- The debounce time in milliseconds
  ipphid_a_to_d_hrul            SMALLINT NOT NULL,                         -- The High Range Upper Limit of A/D ranges
  ipphid_a_to_d_hrll            SMALLINT NOT NULL,                         -- The High Range Lower Limit of A/D ranges
  ipphid_a_to_d_lrul            SMALLINT NOT NULL,                         -- The Low Range Upper Limit of A/D ranges
  ipphid_a_to_d_lrll            SMALLINT NOT NULL,                         -- The Low Range Lower Limit of A/D ranges
  ipphid_report_state_change    SMALLINT NOT NULL,                         -- 1 = Report state changes to host
  site                          CHAR(32) NOT NULL,                         -- Site Name
  ipphid_guid                   uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  ipphid_checksum               INTEGER NOT NULL,                          -- Record checksum
  PRIMARY KEY                   ( ipphid_input_id )
);
```

CREATE TABLE input_hid_save

```
(
  ipphid_input_id               INTEGER NOT NULL,                          -- ID of input
  ipphid_debounce_millisec      SMALLINT NOT NULL,                         -- The debounce time in milliseconds
  ipphid_a_to_d_hrul            SMALLINT NOT NULL,                         -- The High Range Upper Limit of A/D ranges
  ipphid_a_to_d_hrll            SMALLINT NOT NULL,                         -- The High Range Lower Limit of A/D ranges
  ipphid_a_to_d_lrul            SMALLINT NOT NULL,                         -- The Low Range Upper Limit of A/D ranges
  ipphid_a_to_d_lrll            SMALLINT NOT NULL,                         -- The Low Range Lower Limit of A/D ranges
  ipphid_report_state_change    SMALLINT NOT NULL,                         -- 1 = Report state changes to host
  site                          CHAR(32) NOT NULL,                         -- Site Name
  ipphid_guid                   uniqueidentifier NOT NULL,                 -- Global unique record identifier
  ipphid_checksum               INTEGER NOT NULL,                          -- Record checksum
  upd_guid                      uniqueidentifier NOT NULL,                 -- Guid for update operation
```

```
  upd_mode                SMALLINT NOT NULL,                                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp           DATETIME NOT NULL,                                  -- Date / Time of the update
  upd_checksum            INTEGER NOT NULL,                                   -- Checksum
  upd_rowguid             uniqueidentifier ROWGUIDCOL DEFAULT newid()
                            UNIQUE NOT NULL,                                  -- Global unique record identifier
  PRIMARY KEY             ( ipphid_input_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE input_p900

```
(
  ipp_id                  INTEGER REFERENCES input(ip_point_id)
                            NOT FOR REPLICATION,                              -- Unique database ID of input
  ipp_report              SMALLINT NOT NULL,                                  -- report priority -0: none, 1: alarm, 2: transact.
  ipp_monitor_tamper      SMALLINT NOT NULL,                                  -- monitering tamper or not,
  ipp_monitor_input       SMALLINT NOT NULL,                                  -- monitering input or not,
  ipp_shunt_reader        INTEGER NULL REFERENCES terminal(tp_term_id)
                            NOT FOR REPLICATION,                              -- shunt monitoring reader id for this panel
  ipp_shunt_event         SMALLINT,                                           -- shunt event
  site                    CHAR(32) NOT NULL,                                  -- Site Name
  ipp_guid                uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  ipp_checksum            INTEGER NOT NULL,                                   -- Record checksum
  PRIMARY KEY             (ipp_id)
);
```

CREATE TABLE input_p900_save

```
(
  ipp_id                  INTEGER NOT NULL,                                   -- Unique database ID of input
  ipp_report              SMALLINT NOT NULL,                                  -- report priority -0: none, 1: alarm, 2: transact.
  ipp_monitor_tamper      SMALLINT NOT NULL,                                  -- monitering tamper or not,
  ipp_monitor_input       SMALLINT NOT NULL,                                  -- monitering input or not,
  ipp_shunt_reader        INTEGER NULL,                                       -- shunt monitoring reader id for this panel
  ipp_shunt_event         SMALLINT,                                           -- shunt event
  site                    CHAR(32) NOT NULL,                                  -- Site Name
  ipp_guid                uniqueidentifier NOT NULL,                          -- Global unique record identifier
  ipp_checksum            INTEGER NOT NULL,                                   -- Record checksum
  upd_guid                uniqueidentifier NOT NULL,                          -- Guid for update operation
  upd_mode                SMALLINT NOT NULL,                                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp           DATETIME NOT NULL,                                  -- Date / Time of the update
  upd_checksum            INTEGER NOT NULL,                                   -- Checksum
  upd_rowguid             uniqueidentifier ROWGUIDCOL DEFAULT newid()
                            UNIQUE NOT NULL,                                  -- Global unique record identifier
  PRIMARY KEY             (ipp_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE input_s321ip

```
(
  ipps321ip_input_id      INTEGER REFERENCES input(ip_point_id) NOT FOR REPLICATION,  -- ID of input
  ipps321ip_input_debounce SMALLINT NOT NULL,                                 -- The debounce time 40 - 800 msec
  ipps321ip_output_link   SMALLINT NOT NULL,                                  -- Output to follow 0 - 32 (0 = none)
  site                    CHAR(32) NOT NULL,                                  -- Site Name
  ipps321ip_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  ipps321ip_checksum      INTEGER NOT NULL,                                   -- Record checksum
  PRIMARY KEY             ( ipps321ip_input_id )
);
```

CREATE TABLE input_s321ip_save

```
(
  ipps321ip_input_id      INTEGER NOT NULL,                                   -- ID of input
  ipps321ip_input_debounce SMALLINT NOT NULL,                                 -- The debounce time 40 - 800 msec
  ipps321ip_output_link   SMALLINT NOT NULL,                                  -- Output to follow 0 - 32 (0 = none)
```

```
  site                      CHAR(32) NOT NULL,                          -- Site Name
  ipps321ip_guid            uniqueidentifier NOT NULL,                  -- Global unique record identifier
  ipps321ip_checksum        INTEGER NOT NULL,                           -- Record checksum
  upd_guid                  uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode                  SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp             DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum              INTEGER NOT NULL,                           -- Checksum
  upd_rowguid               uniqueidentifier ROWGUIDCOL DEFAULT newid()
                              UNIQUE NOT NULL,                          -- Global unique record identifier
  PRIMARY KEY               ( ipps321ip_input_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE input_save

```
(
  ip_partition              INTEGER NOT NULL,                           -- ID of partition for this input
  ip_public                 SMALLINT,                                   -- publication
  ip_point_id               INTEGER NOT NULL,                           -- Unique database ID of input point
  ip_point_name             varchar(32),                                -- name of input point
  ip_point_number           SMALLINT,                                   -- input number
  ip_term_id                INTEGER NULL,                               -- terminal ID that has this input, is null if this is a panel soft
                                                                           input,  if null then ip_panel_id is used
  ip_panel_id               INTEGER NULL,                               -- panel ID of this input, is used only for panel soft inputs
                                                                        -- this is null if ip_term_id is used-- only 1 of these fields
                                                                        -- can be used for any given input point
  ip_state                  SMALLINT,                                   -- input state
  ip_enable_flag            SMALLINT,                                   -- input enable
  ip_query_string           CHAR(64),                                   -- query string enter by user
  ip_soft_input             SMALLINT,                                   -- 1 = soft input, 0 = hardware input
  ip_suppress_tz            INTEGER NULL,                               -- timezone ID when input suppressed (always active if NULL)
  site                      CHAR(32) NOT NULL,                          -- Site Name
  ip_guid                   uniqueidentifier NOT NULL,                  -- Global unique record identifier
  ip_checksum               INTEGER NOT NULL,                           -- Record checksum -- ip_disp_alarm is valid
  upd_guid                  uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode                  SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp             DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum              INTEGER NOT NULL,                           -- Checksum
  upd_rowguid               uniqueidentifier ROWGUIDCOL DEFAULT newid()
                              UNIQUE NOT NULL,                          -- Global unique record identifier
  PRIMARY KEY               ( ip_partition, ip_point_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE IntegrationComponent

```
(
  igc_name                  VARCHAR(32) NOT NULL,                       -- user entered name
  igc_partition             INTEGER REFERENCES partition(part_number)
                              NOT FOR REPLICATION,                      -- partition ID to which this integration component belongs
  igc_public                SMALLINT,                                   -- 1 = public, 0 = private
  igc_id                    INTEGER IDENTITY NOT FOR
                              REPLICATION UNIQUE,                       -- unique database ID
  igc_query_string          CHAR(64) NULL,                              -- query string entered by user
  igc_type                  SMALLINT NOT NULL,                          -- IntegrationComponent type
  igc_set_children_when_down SMALLINT NOT NULL,                         -- 0 = no change, 1 = set down, 2 = set unknown
  igc_ip_addr               CHAR(16) NULL,                              -- IP address, if used
  igc_computer_name         CHAR(64) NULL,                              -- Computer Name, if used
  igc_port                  INT NULL,                                   -- Network port number, if used
  igc_comms_port            INT NULL,                                   -- RS232 /485 Comms port number, if used
  igc_comms_parameters      VARCHAR(32) NULL,                           -- RS232 /485 Comms parameters, if used
  site                      CHAR(32) NOT NULL,                          -- Site Name
  igc_guid                  uniqueidentifier ROWGUIDCOL UNIQUE
                              NOT NULL,                                 -- Global unique record identifier
  igc_checksum              INTEGER NOT NULL,                           -- Record checksum
  igc_username              VARCHAR(32) NULL,                           -- password for connection
```

```
  igc_password_0          INTEGER NULL,                                   -- encrypted password data
  igc_password_1          INTEGER NULL,                                   -- encrypted password data
  igc_password_2          INTEGER NULL,                                   -- encrypted password data
  igc_password_3          INTEGER NULL,                                   -- encrypted password data
  PRIMARY KEY             ( igc_id )
);
  CREATE INDEX            igc_name ON IntegrationComponent( igc_name );
  CREATE UNIQUE INDEX     igc_site_name ON IntegrationComponent( site, igc_name );
```

CREATE TABLE IntegrationComponentStatus

```
(
  igcs_id                 INTEGER REFERENCES IntegrationComponent(igc_id)
                            NOT FOR REPLICATION,
  igcs_status             SMALLINT NOT NULL,                              -- Set by related comms (see IntegrationComponentsStatus.h)
  igcs_reported_version   VARCHAR(32) NULL,                               -- Reported version string
  igcs_status_time        DATETIME NOT NULL,                              -- When status set
  igcs_last_comms_time    DATETIME NULL,                                  -- Last communications time
  site                    CHAR(32) NOT NULL,                              -- Site Name
  igcs_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
  igcs_checksum           INTEGER NOT NULL,                               -- Record checksum
  PRIMARY KEY             ( igcs_id )
);
```

CREATE TABLE IntegrationComponent_assaabloy_dsr

```
(
  [dsr_igc_id]            INTEGER REFERENCES IntegrationComponent(igc_id) NOT FOR REPLICATION NOT NULL,
  [dsr_encryption]        SMALLINT NULL,
  site                    CHAR(32) NOT NULL,
  [dsr_guid]              [uniqueidentifier] ROWGUIDCOL PRIMARY KEY DEFAULT newid() NOT NULL,
  [dsr_checksum]          [int] NOT NULL,
)
```

CREATE TABLE IntegrationComponent_assaabloy_dsr_save

```
(
  dsr_igc_id              INTEGER NOT NULL,
  dsr_encryption          SMALLINT NULL,
  site                    CHAR(32) NOT NULL,
  dsr_guid                uniqueidentifier NOT NULL,
  dsr_checksum            INTEGER NOT NULL,
  upd_guid                uniqueidentifier NOT NULL, -                    - Guid for update operation
  upd_mode                SMALLINT NOT NULL,                              -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp           DATETIME NOT NULL,                              -- Date / Time of the update
  upd_checksum            INTEGER NOT NULL,                               -- Checksum
  upd_rowguid             uniqueidentifier ROWGUIDCOL DEFAULT newid()
                            UNIQUE NOT NULL,                              -- Global unique record identifier
  PRIMARY KEY             ( dsr_guid, upd_mode desc, upd_guid )
)
```

CREATE TABLE IntegrationComponent_intercom

```
(
  igic_integrationcomponent_guid  uniqueidentifier REFERENCES IntegrationComponent(igc_guid) NOT FOR REPLICATION,
  igic_enable_alarm       smallint NOT NULL,                              -- enable importing alarms to P2000, disable = 0/enable = 1
  igic_save_xaction       smallint NOT NULL,                              -- yes=1; no=0
  igic_mfg_id             INTEGER REFERENCES msgfiltergroup(mfg_id)
                            NOT FOR REPLICATION,                          -- message filter group ID
  igic_dial_length        INTEGER NOT NULL,                               -- max munber of dialing digits used in this interface
  igic_support_regions    smallint NOT NULL,                              -- support=1; no support=0;
  igic_region             char(32) NULL,
  igic_guid               uniqueidentifier ROWGUIDCOL NOT NULL,
  igic_checksum           INTEGER NOT NULL,
```

```
  PRIMARY KEY                 (igic_guid)
)
```

CREATE TABLE IntegrationComponent_intercom_save

```
(
  igic_integrationcomponent_guid  uniqueidentifier NULL,
  igic_enable_alarm               smallint NOT NULL,          -- enable importing alarms to P2000, disable = 0/enable = 1
  igic_save_xaction               smallint NOT NULL,          -- yes=1; no=0
  igic_mfg_id                     INTEGER NOT NULL,           -- message filter group ID
  igic_dial_length                INTEGER NOT NULL,           -- max munber of dialing digits used in this interface
  igic_support_regions            smallint NOT NULL,          -- support=1; no support=0;
  igic_region                     char(32) NULL,
  igic_guid                       uniqueidentifier NOT NULL,
  igic_checksum                   INTEGER NOT NULL,
  upd_guid                        uniqueidentifier NOT NULL,  -- Guid for update operation
  upd_mode                        SMALLINT NOT NULL,          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                   DATETIME NOT NULL,          -- Date / Time of the update
  upd_checksum                    INTEGER NOT NULL,           -- Checksum
  upd_rowguid                     uniqueidentifier ROWGUIDCOL DEFAULT
                                    newid() UNIQUE NOT NULL,  -- Global unique record identifier
  PRIMARY KEY                     (igic_guid, upd_mode desc, upd_guid )
)
```

CREATE TABLE IntegrationComponent_save

```
(
  igc_name                    VARCHAR(32) NOT NULL,   -- user entered name
  igc_partition               INTEGER,                -- partition ID to which this integration component belongs
  igc_public                  SMALLINT,               -- 1 = public, 0 = private
  igc_id                      INTEGER NOT NULL,       -- unique database ID
  igc_query_string            CHAR(64) NULL,          -- query string entered by user
  igc_type                    SMALLINT NOT NULL,      -- IntegrationComponent type
  igc_set_children_when_down  SMALLINT NOT NULL,      -- 0 = no change, 1 = set down, 2 = set unknown
  igc_ip_addr                 CHAR(16) NULL,          -- IP address if used
  igc_computer_name           CHAR(64) NULL,          -- Computer Name, if used
  igc_port                    INT NULL,               -- Network port number if used
  igc_comms_port              INT NULL,               -- RS232 /485 Comms port number if used
  igc_comms_parameters        VARCHAR(32) NULL,       -- RS232 /485 Comms parameters, component specific if used
  site                        CHAR(32) NOT NULL,      -- Site Name
  igc_guid                    uniqueidentifier NOT NULL,  -- Global unique record identifier
  igc_checksum                INTEGER NOT NULL,       -- Record checksum
  upd_guid                    uniqueidentifier NOT NULL,  -- Guid for update operation
  upd_mode                    SMALLINT NOT NULL,      -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp               DATETIME NOT NULL,      -- Date / Time of the update
  upd_checksum                INTEGER NOT NULL,       -- Checksum
  upd_rowguid                 uniqueidentifier ROWGUIDCOL DEFAULT
                                newid() UNIQUE NOT NULL,  -- Global unique record identifier
  igc_username                VARCHAR(32) NULL,       -- password for connection
  igc_password_0              INTEGER NULL,           -- encrypted password data
  igc_password_1              INTEGER NULL,           -- encrypted password data
  igc_password_2              INTEGER NULL,           -- encrypted password data
  igc_password_3              INTEGER NULL,           -- encrypted password data
  PRIMARY KEY                 ( igc_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE intercom_exchange

```
(
  inte_partition  INTEGER REFERENCES partition(part_number)
                    NOT FOR REPLICATION,                  -- partition ID to which this exchange belongs
  inte_public     SMALLINT,                               -- 1 = exchange is public, 0 = exchange is private
  inte_id         INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,  -- unique database ID
  inte_name       char(32) NOT NULL,                      -- exchange name
```

```
  inte_query_string       char(64) NULL,
  inte_device_type        char(255) NULL,
  inte_version            int NOT NULL,
  inte_interface_id       int REFERENCES intercom_interface(inti_id)
                            NOT FOR REPLICATION,
  inte_region             char(32) NULL,
  site                    CHAR(32) NOT NULL,                          -- Site Name
  inte_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  inte_checksum           INTEGER NOT NULL,                           -- Record checksum
  PRIMARY KEY             (inte_partition, inte_id )
)
  CREATE UNIQUE INDEX     inte_site_name ON intercom_exchange (site, inte_name);
```

CREATE TABLE intercom_exchange_save (

```
  inte_partition          INTEGER NOT NULL,                           -- partition ID to which this exchange belongs
  inte_public             SMALLINT NOT NULL,                          -- 1 = exchange is public, 0 = exchange is private
  inte_id                 INTEGER NOT NULL,                           -- unique database ID
  inte_name               char(32) NOT NULL,                          -- exchange name
  inte_query_string       char(64) NULL,
  inte_device_type        char(255) NULL,
  inte_version            int NOT NULL,
  inte_interface_id       int NOT NULL,
  inte_region             char(32) NULL,
  site                    CHAR(32) NOT NULL,                          -- Site Name
  inte_guid               uniqueidentifier NOT NULL,                  -- Global unique record identifier
  inte_checksum           INTEGER NOT NULL,                           -- Record checksum
  upd_guid                uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode                SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp           DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum            INTEGER NOT NULL,                           -- Checksum
  upd_rowguid             uniqueidentifier ROWGUIDCOL DEFAULT newid()
                            UNIQUE NOT NULL,                          -- Global unique record identifier
  PRIMARY KEY             (inte_partition, inte_id, upd_mode desc, upd_guid )
)
```

CREATE TABLE intercom_interface(

```
  inti_id                 INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,
  inti_name               char(32) NOT NULL,
  inti_query_string       char(64),
  inti_partition          INTEGER REFERENCES partition(part_number)
                            NOT FOR REPLICATION,
  inti_public             smallint NOT NULL,
  inti_type               INTEGER NOT NULL,                           -- Zenitel MPC protocol - AMC 07.60=1,
                                                                         Commend ICX Version1.1/0910=2
  inti_comms_type         smallint NOT NULL,                          -- Serial Port=1; Ethernet=2
  inti_port               smallint NOT NULL,                          -- Serial Port Number
  inti_baud_rate          INTEGER NOT NULL,                           -- baudrate
  inti_data_bits          smallint NOT NULL,
  inti_parity             smallint NOT NULL,
  inti_stop_bits          smallint NOT NULL,
  inti_server_name        char(32),                                   -- intercom interface host's network name
  inti_ip_address         char(16),                                   -- intercom interface host's network ip address
  inti_ip_port            INTEGER NOT NULL,                           -- ethernet connection port for the interface
  inti_user_name          char(32),                                   -- intercom interface login credential
  inti_password           char(32),                                   -- intercom interface login credential
  inti_save_xaction       smallint NOT NULL,                          -- yes=1; no=0
  inti_mfg_id             INTEGER REFERENCES msgfiltergroup(mfg_id)
                            NOT FOR REPLICATION,                      -- message filter group ID
  inti_dial_length        INTEGER NOT NULL,                           -- max munber of dialing digits used in this interface
  inti_support_regions    smallint NOT NULL,                          -- support=1; no support=0;
  site                    char(32) NOT NULL,
  inti_guid               uniqueidentifier ROWGUIDCOL NOT NULL,
```

```
  inti_checksum          INTEGER NOT NULL,
  PRIMARY KEY            (inti_partition, inti_id)
)
```

CREATE TABLE intercom_interface_save

```
(
  inti_id                INTEGER NOT NULL,
  inti_name              char(32) NOT NULL,
  inti_query_string      char(64),
  inti_partition         INTEGER NOT NULL,
  inti_public            smallint NOT NULL,
  inti_type              INTEGER NOT NULL,                    -- Zenitel MPC protocol - AMC 07.60=1,
                                                                 Commend ICX Version1.1/0910=2
  inti_comms_type        smallint NOT NULL,                   -- Serial Port=1; Ethernet=2
  inti_port              smallint NOT NULL,                   -- Serial Port Number
  inti_baud_rate         INTEGER NOT NULL,                    -- baudrate
  inti_data_bits         smallint NOT NULL,
  inti_parity            smallint NOT NULL,
  inti_stop_bits         smallint NOT NULL,
  inti_server_name       char(32),                            -- intercom interface host's network name
  inti_ip_address        char(16),                            -- intercom interface host's network ip address
  inti_ip_port           INTEGER NOT NULL,                    -- ethernet connection port for the interface
  inti_user_name         char(32),                            -- intercom interface login credential
  inti_password          char(32),                            -- intercom interface login credential
  inti_save_xaction      smallint NOT NULL,                   -- yes=1; no=0
  inti_mfg_id            INTEGER NOT NULL,                    -- message filter group ID
  inti_dial_length       INTEGER NOT NULL,                    -- max munber of dialing digits used in this interface
  inti_support_regions   smallint NOT NULL,                   -- support=1; no support=0;
  site                   char(32) NOT NULL,
  inti_guid              uniqueidentifier NOT NULL,
  inti_checksum          INTEGER NOT NULL,
  upd_guid               uniqueidentifier NOT NULL,           -- Guid for update operation
  upd_mode               SMALLINT NOT NULL,                   -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp          DATETIME NOT NULL,                   -- Date / Time of the update
  upd_checksum           INTEGER NOT NULL,                    -- Checksum
  upd_rowguid            uniqueidentifier ROWGUIDCOL DEFAULT newid()
                             UNIQUE NOT NULL,                 -- Global unique record identifier
  PRIMARY KEY            (inti_partition, inti_id, upd_mode desc, upd_guid )
)
```

CREATE TABLE intercom_station

```
(
  ic_stn_partition                INTEGER REFERENCES partition(part_number)
                                     NOT FOR REPLICATION,      -- partition ID to which this station belongs
  ic_stn_public                   SMALLINT NOT NULL,           -- 1 = station is public, 0 = station is private
  ic_stn_query_string             char(64) NULL,
  ic_stn_id                       INTEGER IDENTITY NOT FOR REPLICATION,   -- unique database ID
  ic_stn_address                  INTEGER,                     -- unique station ID
  ic_stn_name                     char(32) NOT NULL,           -- station name
  ic_stn_priority                 SMALLINT NOT NULL,           -- station defined priority
  ic_stn_type                     SMALLINT NOT NULL,           -- station defined type
  ic_stn_workstation_id           INTEGER references station(stn_id) NOT FOR REPLICATION,   -- Workstation ID
  site                            CHAR(32) NOT NULL,           -- Site Name
  ic_stn_guid                     uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
  ic_stn_checksum                 INTEGER NOT NULL,            -- Record checksum
  ic_stn_integrationcomponent_id  INTEGER references IntegrationComponent(igc_id)
                                     NOT FOR REPLICATION,      -- IntegrationComponent ID which this
                                                                  station belongs to
  ic_stn_ref_guid                 uniqueidentifier NULL,       -- reference to station or integration
                                                                  component
  PRIMARY KEY                     (ic_stn_id )
)
```

CREATE TABLE intercom_station_status

```
(
  icstat_id              INTEGER REFERENCES intercom_station(ic_stn_id)
                           NOT FOR REPLICATION,                    -- unique station ID
  icstat_status          SMALLINT,                                 -- current reader status 0-5 (set by comms)
  icstat_timestamp       DATETIME,                                 -- current reader status time (set by comms)
  site                   CHAR(32) NOT NULL,                        -- Site Name
  icstat_guid            uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  icstat_checksum        INTEGER NOT NULL,                         -- Record checksum
  PRIMARY KEY            (icstat_id)
);
```

CREATE TABLE intercom_station_save

```
(
  ic_stn_partition                INTEGER,                         -- partition ID to which this station belongs
  ic_stn_public                   SMALLINT,                        -- 1 = station is public, 0 = station is private
  ic_stn_query_string             char(64) NULL,
  ic_stn_id                       INTEGER,                         -- unique database ID
  ic_stn_address                  INTEGER,                         -- unique station ID
  ic_stn_name                     char(32) NOT NULL,               -- station name
  ic_stn_priority                 SMALLINT NOT NULL,               -- station defined priority
  ic_stn_type                     SMALLINT NOT NULL,               -- station defined type
  ic_stn_workstation_id           INTEGER,                         -- Workstation ID
  site                            CHAR(32) NOT NULL,               -- Site Name
  ic_stn_guid                     uniqueidentifier NOT NULL,       -- Global unique record identifier
  ic_stn_checksum                 INTEGER NOT NULL,                -- Record checksum
  upd_guid                        uniqueidentifier NOT NULL,       -- Guid for update operation
  upd_mode                        SMALLINT NOT NULL,               -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                   DATETIME NOT NULL,               -- Date / Time of the update
  upd_checksum                    INTEGER NOT NULL,                -- Checksum
  upd_rowguid                     uniqueidentifier ROWGUIDCOL DEFAULT
                                    newid() UNIQUE NOT NULL,       -- Global unique record identifier
  ic_stn_integrationcomponent_id  INTEGER,                         -- IntegrationComponent ID which this station
                                                                      belongs to
  ic_stn_ref_guid                 uniqueidentifier NULL,           -- reference to station or integration component
  PRIMARY KEY                     (ic_stn_id, upd_mode DESC, upd_guid )
)
```

CREATE TABLE intrusion_entity

```
(
  intr_id int            IDENTITY NOT FOR REPLICATION unique,
  intr_type              int not NULL ,
  intr_partition         INTEGER REFERENCES partition(part_number)
                           NOT FOR REPLICATION,                    -- ID of partition for this input
  intr_public            smallint NULL ,
  intr_itemid            int NOT NULL,
  intr_name              varchar(32) NOT NULL ,
  intr_opcname           varchar(255) ,
  intr_parentid          int NULL ,
  site                   CHAR(32) NOT NULL,                        -- Site Name
  intr_querystring       varchar(64),
  intr_enable            smallint,
  intr_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  intr_checksum          int NOT NULL,
  intr_integration_id    INTEGER REFERENCES IntegrationComponent(igc_id)
                           NOT FOR REPLICATION,
  PRIMARY KEY            ( intr_partition, intr_id)
)
```

CREATE TABLE intrusion_entity_save

```
(
  intr_id              INTEGER NOT NULL,                          -- Unique database ID of intrusion entity
  intr_type            INTEGER NOT NULL ,                         -- Intrusion Entity Type
  intr_partition       INTEGER NOT NULL ,
  intr_public          smallint NOT NULL ,
  intr_itemid          int NOT NULL,
  intr_name            char(32) ,
  intr_opcname         char(255) ,
  intr_parentid        int NULL ,
  intr_guid            uniqueidentifier NOT NULL ,
  intr_checksum        int NOT NULL,
  site                 CHAR(32) NOT NULL,                         -- Site Name
  intr_querystring     varchar(64),
  intr_enable          smallint,
  upd_guid             uniqueidentifier NOT NULL,                 -- Guid for update operation
  upd_mode             SMALLINT NOT NULL,                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp        DATETIME NOT NULL,                         -- Date / Time of the update
  upd_checksum         INTEGER NOT NULL,                          -- Checksum
  upd_rowguid          uniqueidentifier ROWGUIDCOL DEFAULT newid()
                         UNIQUE NOT NULL,                         -- Global unique record identifier
  intr_integration_id  INTEGER,
  PRIMARY KEY          (intr_id, upd_mode desc, upd_guid)
)
```

CREATE TABLE intrusion_status

```
(
  intrstat_id                      INTEGER REFERENCES intrusion_entity(intr_id)
                                     NOT FOR REPLICATION,         -- Intrusion ID to which this status belongs
  intrstat_status                  SMALLINT NOT NULL,             -- Icon status
  intrstat_substatus               SMALLINT NOT NULL,             -- SubStatus
  intrstat_timestamp               DATETIME NOT NULL,             -- time of last current state (set by comms)
  site                             CHAR(32) NOT NULL,             -- Site Name
  intrstat_guid                    uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  intrstat_checksum                INTEGER NOT NULL,              -- Record checksum
  intrstat_prev_event_log_ent      INTEGER,                       -- For Bosch Intrusion
  intrstat_last_block_event_counter  INTEGER,                     -- For Bosch Intrusion
  intrstat_first_block             VARBINARY(100),                -- For Bosch Intrusion
  PRIMARY KEY                      (intrstat_id)
);
```

CREATE TABLE itemCategory

```
(
  itemReference        VARCHAR (1024) NOT NULL,                   -- MSEA item fully qualified reference name
  itemValue            VARCHAR (50) NOT NULL                      -- MSEA item object category property value
);
```

CREATE TABLE journal

```
(
  site                 CHAR(32) NOT NULL,                         -- Site Name
  journal_guid         uniqueidentifier DEFAULT newid() ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  journal_owner_guid   uniqueidentifier NOT NULL,                 -- database GUID of owning item
  journal_title        VARCHAR(64) NOT NULL,                      -- title of journal entry
  journal_entry        TEXT,                                      -- Text of the journal entry
  journal_create_time  DATETIME NOT NULL,                         -- Time this journal was created
  journal_edit_time    DATETIME,                                  -- Time this journal was last edited
  journal_checksum     INTEGER NOT NULL,                          -- Record checksum
  journal_operator     CHAR(32),                                  -- Operator that last modified record
  PRIMARY KEY          NONCLUSTERED ( journal_guid )
);
  CREATE CLUSTERED INDEX     journal_owner_guid ON journal(journal_owner_guid);
```

```
CREATE INDEX        journal_create_time ON journal(journal_create_time);
CREATE INDEX        journal_title ON journal(journal_title);
```

CREATE TABLE journal_attachments

```
(
  ja_journal_guid       uniqueidentifier REFERENCES journal(journal_guid),   -- journal entry identifier
  ja_filename           VARCHAR(255),                                        -- original filename
  ja_file_data          IMAGE,                                               -- attachment file data
  site                  CHAR(32) NOT NULL,                                   -- Site Name
  ja_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,         -- Global unique record identifier
  ja_checksum           INTEGER NOT NULL,                                    -- Record checksum
  PRIMARY KEY           ( ja_guid )
);
CREATE INDEX            ja_journal_guid ON journal(journal_guid);
```

CREATE TABLE journal_attachments_save

```
(
  ja_journal_guid       uniqueidentifier,                                    -- journal entry identifier
  ja_filename           VARCHAR(255),                                        -- original filename
  ja_file_data          IMAGE,                                               -- attachment file data
  site                  CHAR(32) NOT NULL,                                   -- Site Name
  ja_guid               uniqueidentifier NOT NULL,                           -- Global unique record identifier
  ja_checksum           INTEGER NOT NULL,                                    -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                           -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                   -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                   -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                    -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,                           -- Global unique record identifier
  PRIMARY KEY           ( ja_guid, upd_mode desc, upd_guid )
);
```

CREATE TABLE journal_save

```
(
  site                  CHAR(32) NOT NULL,                                   -- Site Name
  journal_guid          uniqueidentifier NOT NULL,                           -- Global unique record identifier
  journal_owner_guid    uniqueidentifier NOT NULL,                           -- database GUID of owning item
  journal_title         VARCHAR(64) NOT NULL,                                -- title of journal entry
  journal_entry         TEXT,                                                -- Text of the journal entry
  journal_create_time   DATETIME NOT NULL,                                   -- Time this journal was created
  journal_edit_time     DATETIME,                                            -- Time this journal was last edited
  journal_checksum      INTEGER NOT NULL,                                    -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                           -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                   -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                   -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                    -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,  -- Global unique record identifier
  journal_operator      CHAR(32),                                            -- Operator that last modified record
  PRIMARY KEY           NONCLUSTERED ( journal_guid, upd_guid )
);
```

CREATE TABLE koneip_cntrl_defn

```
(
  kipcd_id              INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,         -- unique database ID
  kipcd_panel_id        INTEGER REFERENCES panel(p_panel_id)
                          NOT FOR REPLICATION,                               -- A CK72x panel reference
  kipcd_cntrl_type      SMALLINT NOT NULL,                                   -- type 0 = KIC, 1 = Primary / Backup
  kipcd_primary_name    CHAR(32) NOT NULL,                                   -- user assigned name
  kipcd_primary_ip_addr CHAR(16) NULL,                                       -- ip address of kone controller
  kipcd_backup_name     CHAR(32) NULL,                                       -- user assigned name
```

```
  kipcd_backup_ip_addr    CHAR(16) NULL,                                  -- ip address of kone controller
  kipcd_tcp_port          INT NOT NULL,                                   -- tcp port (0- 65535)
  kipcd_send_cop          SMALLINT NOT NULL,                              -- 0 = false, 1 = true
  kipcd_send_dop          SMALLINT NOT NULL,                              -- 0 = false, 1 = true
  kipcd_hb_interval       SMALLINT NOT NULL,                              -- heart beat interval (5 - 15 seconds)
  site                    CHAR(32) NOT NULL,                              -- Site Name
  kipcd_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
  kipcd_checksum          INTEGER NOT NULL,                               -- Record checksum
  PRIMARY KEY             (kipcd_id, kipcd_panel_id, site)
);
```

CREATE TABLE koneip_cntrl_defn_save

```
(
  kipcd_id                INTEGER NOT NULL,                               -- unique identifier
  kipcd_panel_id          INTEGER NOT NULL,                               -- A CK72x panel reference
  kipcd_cntrl_type        SMALLINT NOT NULL,                              -- type 0 = KIC, 1 = Primary, 2 = Backup
  kipcd_primary_name      CHAR(32) NOT NULL,                              -- user assigned name
  kipcd_primary_ip_addr   CHAR(16) NULL,                                  -- ip address of kone controller
  kipcd_backup_name       CHAR(32) NULL,                                  -- user assigned name
  kipcd_backup_ip_addr    CHAR(16) NULL,                                  -- ip address of kone controller
  kipcd_tcp_port          INT NOT NULL,                                   -- tcp port (0- 65535)
  kipcd_send_cop          SMALLINT NOT NULL,                              -- 0 = false, 1 = true
  kipcd_send_dop          SMALLINT NOT NULL,                              -- 0 = false, 1 = true
  kipcd_hb_interval       SMALLINT NOT NULL,                              -- heart beat interval (5 - 15 seconds)
  site                    CHAR(32) NOT NULL,                              -- Site Name
  kipcd_guid              uniqueidentifier NOT NULL,                      -- Global unique record identifier
  kipcd_checksum          INTEGER NOT NULL,                               -- Record checksum
  upd_guid                uniqueidentifier NOT NULL,                      -- Guid for update operation
  upd_mode                SMALLINT NOT NULL,                              -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp           DATETIME NOT NULL,                              -- Date / Time of the update
  upd_checksum            INTEGER NOT NULL,                               -- Checksum
  upd_rowguid             uniqueidentifier ROWGUIDCOL DEFAULT newid()
                            UNIQUE NOT NULL,                              -- Global unique record identifier
  PRIMARY KEY             (kipcd_id, kipcd_panel_id, site, upd_mode desc, upd_guid)
);
```

CREATE TABLE koneip_cntrl_group_item

```
(
  kipcgi_cntrl_id         INTEGER REFERENCES koneip_cntrl_defn(kipcd_id)
                            NOT FOR REPLICATION,                          -- A CK72x panel reference
  kipcgi_number           SMALLINT NOT NULL,                              -- 0 - 32 used to place entries in UI and sort
  kipcgi_tcp_port         INT NOT NULL,                                   -- tcp port (0- 65535)
  kipcgi_group_addr       SMALLINT NOT NULL,                              -- group (0 - 32)
  site                    CHAR(32) NOT NULL,                              -- Site Name
  kipcgi_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
  kipcgi_checksum         INTEGER NOT NULL,                               -- Record checksum
  PRIMARY KEY             (kipcgi_cntrl_id, kipcgi_number, site)
);
```

CREATE TABLE koneip_cntrl_group_item_save

```
(
  kipcgi_cntrl_id         INTEGER NOT NULL,                               -- Matching panel number
  kipcgi_number           SMALLINT NOT NULL,                              -- 0 - 8 used to place entries in UI and sort
  kipcgi_tcp_port         INT NOT NULL,                                   -- tcp port (0- 65535)
  kipcgi_group_addr       SMALLINT NOT NULL,                              -- group (0 - 32)
  site                    CHAR(32) NOT NULL,                              -- Site Name
  kipcgi_guid             uniqueidentifier NOT NULL,                      -- Global unique record identifier
  kipcgi_checksum         INTEGER NOT NULL,                               -- Record checksum
  upd_guid                uniqueidentifier NOT NULL,                      -- Guid for update operation
  upd_mode                SMALLINT NOT NULL,                              -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp           DATETIME NOT NULL,                              -- Date / Time of the update
```

```
  upd_checksum          INTEGER NOT NULL,                                  -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT
                          newid() UNIQUE NOT NULL,                         -- Global unique record identifier
  PRIMARY KEY           (kipcgi_cntrl_id, kipcgi_number, site, upd_mode desc, upd_guid)
);
```

CREATE TABLE koneip_cntrl_grp

```
(
  kipcg_id              INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,       -- unique database ID
  kipcg_cntrl_id        INTEGER REFERENCES koneip_cntrl_defn(kipcd_id)
                          NOT FOR REPLICATION,                             -- Kone IP controller reference
  kipcg_group_number    SMALLINT NOT NULL,                                 -- 0 - 32 used to identify group number
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  kipcg_guid            uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  kipcg_checksum        INTEGER NOT NULL,                                  -- Record checksum
  PRIMARY KEY           (kipcg_id, kipcg_cntrl_id, site)
);
```

CREATE TABLE koneip_cntrl_grp_floor

```
(
  kipcgf_id             INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,       -- unique database ID
  kipcgf_cntrl_grp_id   INTEGER REFERENCES koneip_cntrl_grp(kipcg_id)
                          NOT FOR REPLICATION,                             -- Kone IP Group reference
  kipcgf_floor          SMALLINT NOT NULL,                                 -- Floor number (1 - 128)
  kipcgf_level          SMALLINT NOT NULL,                                 -- Level 0 - 128 (0 = not defined)
  kipcgf_side           SMALLINT NOT NULL,                                 -- 0 = both, 1 = front, 2 = rear
  kipcgf_cop_dst_connect    SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipcgf_cop_dst_disconnect SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipcgf_dop_dst_connect    SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipcgf_dop_dst_disconnect SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipcgf_dop_src_connect    SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipcgf_dop_src_disconnect SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  kipcgf_guid           uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  kipcgf_checksum       INTEGER NOT NULL,                                  -- Record checksum
  PRIMARY KEY           (kipcgf_id, kipcgf_cntrl_grp_id, site)
);
```

CREATE TABLE koneip_cntrl_grp_floor_save

```
(
  kipcgf_id             INTEGER NOT NULL,                                  -- unique database ID
  kipcgf_cntrl_grp_id   INTEGER NOT NULL,                                  -- Kone IP Group reference
  kipcgf_floor          SMALLINT NOT NULL,                                 -- Floor number (1 - 128)
  kipcgf_level          SMALLINT NOT NULL,                                 -- Level 0 - 128 (0 = not defined)
  kipcgf_side           SMALLINT NOT NULL,                                 -- 0 = both, 1 = front, 2 = rear
  kipcgf_cop_dst_connect    SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipcgf_cop_dst_disconnect  SMALLINT NOT NULL,                            -- 0 = false, 1 = true
  kipcgf_dop_dst_connect    SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipcgf_dop_dst_disconnect SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipcgf_dop_src_connect    SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipcgf_dop_src_disconnect SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  kipcgf_guid           uniqueidentifier NOT NULL,                         -- Global unique record identifier
  kipcgf_checksum       INTEGER NOT NULL,                                  -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                         -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                 -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                  -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                 -- Global unique record identifier
  PRIMARY KEY           (kipcgf_id, kipcgf_cntrl_grp_id, site, upd_mode desc, upd_guid)
);
```

CREATE TABLE koneip_cntrl_grp_save

```
(
  kipcg_id              INTEGER NOT NULL,                                  -- unique database ID
  kipcg_cntrl_id        INTEGER NOT NULL,                                  -- Kone IP controller reference
  kipcg_group_number    SMALLINT NOT NULL,                                 -- 0 - 32 used to identify group number
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  kipcg_guid            uniqueidentifier NOT NULL,                         -- Global unique record identifier
  kipcg_checksum        INTEGER NOT NULL,                                  -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                         -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                 -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                  -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                 -- Global unique record identifier
  PRIMARY KEY           (kipcg_id, kipcg_cntrl_id, site, upd_mode desc, upd_guid)
);
```

CREATE TABLE koneip_floor_cfg

```
(
  kipfc_id                  INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,   -- unique database ID
  kipfc_public              SMALLINT,                                      -- 1 = public, 0 = private
  kipfc_floor               SMALLINT NOT NULL,                             -- Floor number (1 - 128)
  kipfc_level               SMALLINT NOT NULL,                             -- Level 0 - 128
  kipfc_side                SMALLINT NOT NULL,                             -- 0 = both, 1 = front, 2 = rear
  kipfc_cop_dst_connect     SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipfc_cop_dst_disconnect  SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipfc_dop_dst_connect     SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipfc_dop_dst_disconnect  SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipfc_dop_src_connect     SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipfc_dop_src_disconnect  SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  site                      CHAR(32) NOT NULL,                             -- Site Name
  kipfc_guid                uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
  kipfc_checksum            INTEGER NOT NULL,                              -- Record checksum
  PRIMARY KEY               (kipfc_id)
);
```

CREATE TABLE koneip_floor_cfg_save

```
(
  kipfc_id                  INTEGER NOT NULL,                              -- unique database ID
  kipfc_public              SMALLINT,                                      -- 1 = public, 0 = private
  kipfc_floor               SMALLINT NOT NULL,                             -- Floor number (1 - 128)
  kipfc_level               SMALLINT NOT NULL,                             -- Level 0 - 128
  kipfc_side                SMALLINT NOT NULL,                             -- 0 = both, 1 = front, 2 = rear
  kipfc_cop_dst_connect     SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipfc_cop_dst_disconnect  SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipfc_dop_dst_connect     SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipfc_dop_dst_disconnect  SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipfc_dop_src_connect     SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  kipfc_dop_src_disconnect  SMALLINT NOT NULL,                             -- 0 = false, 1 = true
  site                      CHAR(32) NOT NULL,                             -- Site Name
  kipfc_guid                uniqueidentifier NOT NULL,                     -- Global unique record identifier
  kipfc_checksum            INTEGER NOT NULL,                              -- Record checksum
  upd_guid                  uniqueidentifier NOT NULL,                     -- Guid for update operation
  upd_mode                  SMALLINT NOT NULL,                             -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp             DATETIME NOT NULL,                             -- Date / Time of the update
  upd_checksum              INTEGER NOT NULL,                              -- Checksum
  upd_rowguid               uniqueidentifier ROWGUIDCOL DEFAULT newid()
                              UNIQUE NOT NULL,                             -- Global unique record identifier
  PRIMARY KEY               (kipfc_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE localsite

```
(
  ls_site_name          CHAR(32) UNIQUE NOT NULL,                          -- local site name
  ls_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  ls_checksum           INTEGER NOT NULL,                                  -- Record checksum
);
```

CREATE TABLE localsite_save

```
(
  ls_site_name          CHAR(32) NOT NULL,                                 --  local site name
  ls_guid               uniqueidentifier NOT NULL,                         -- Global unique record identifier
  ls_checksum           INTEGER NOT NULL,                                  -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                         -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                 -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                  -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,  -- Global unique record identifier
  PRIMARY KEY           (upd_mode desc, upd_guid)
);
```

CREATE TABLE loopconfig

```
(
  lc_id                 INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,       -- Unique loop ID
  lc_loop_number        SMALLINT NOT NULL,                                 -- Loop number (1 - 32)
  lc_baud_rate          INT NOT NULL,                                      -- Baud rate (the actual number: 9600, 4800, 2400, 1200)
  lc_com_port           SMALLINT NOT NULL,                                 -- 1 = COM1:, ... n == COMn:
  lc_enable             SMALLINT NOT NULL,                                 -- 1 = loop enabled, 0 = disabled
  lc_type               SMALLINT NOT NULL,
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  lc_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  lc_checksum           INTEGER NOT NULL,                                  -- Record checksum
  lc_monitor_tamper     SMALLINT NOT NULL,                                 -- Montior the loop tamper alarm on this loop
);
  CREATE UNIQUE INDEX   lc_site_loop ON loopconfig ( site,lc_loop_number );
  CREATE INDEX          lc_loop_ID ON loopconfig ( lc_id );
```

CREATE TABLE loopconfig_save

```
(
  lc_id                 INTEGER  NOT NULL,                                 -- Unique loop ID
  lc_loop_number        SMALLINT NOT NULL,                                 -- Loop number (1 - 32)
  lc_baud_rate          INT NOT NULL,                                      -- Baud rate (the actual number: 9600, 4800, 2400, 1200)
  lc_com_port           SMALLINT NOT NULL,                                 -- 1 = COM1:, ... n == COMn:
  lc_enable             SMALLINT NOT NULL,                                 -- 1 = loop enabled, 0 = disabled
  lc_type               SMALLINT NOT NULL,
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  lc_guid               uniqueidentifier NOT NULL,                         -- Global unique record identifier
  lc_checksum           INTEGER NOT NULL,                                  -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                         -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                 -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                  -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                 -- Global unique record identifier
  lc_monitor_tamper     SMALLINT NOT NULL,                                 -- Montior the loop tamper alarm on this loop
  PRIMARY KEY           (lc_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE map

```
(
  map_partition        INTEGER REFERENCES partition(part_number)
                         NOT FOR REPLICATION,                    -- partition ID to which this map belongs
  map_public           SMALLINT,                                 -- 1 = public, 0 = private
  map_id               INTEGER IDENTITY NOT FOR REPLICATION UNIQUE, -- unique ID for this map
  map_name             CHAR(32) NOT NULL,                        -- map name
  map_type             SMALLINT,                                 -- 1 = top level map, 2 = normal map, 3, = popup map
  site                 CHAR(32) NOT NULL,                        -- Site Name
  map_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  map_checksum         INTEGER NOT NULL,                         -- Record checksum
  PRIMARY KEY          ( map_partition, map_id )
);
  CREATE INDEX         map_public ON map ( map_public );
  CREATE INDEX         map_type ON map ( map_type );
```

CREATE TABLE mapimage

```
(
  map_id               INTEGER REFERENCES map(map_id) NOT FOR REPLICATION,  -- map ID for this image
  map_data             IMAGE,                                    -- image data
  site                 CHAR(32) NOT NULL,                        -- Site Name
  map_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  map_checksum         INTEGER NOT NULL,                         -- Record checksum
  PRIMARY KEY          (map_id)
);
```

CREATE TABLE mapimage_save

```
(
  map_id               INTEGER NOT NULL,                         -- map ID for this image
  map_data             IMAGE,                                    -- image data
  site                 CHAR(32) NOT NULL,                        -- Site Name
  map_guid             uniqueidentifier NOT NULL,                -- Global unique record identifier
  map_checksum         INTEGER NOT NULL,                         -- Record checksum
  upd_guid             uniqueidentifier NOT NULL,                -- Guid for update operation
  upd_mode             SMALLINT NOT NULL,                        -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp        DATETIME NOT NULL,                        -- Date / Time of the update
  upd_checksum         INTEGER NOT NULL,                         -- Checksum
  upd_rowguid          uniqueidentifier ROWGUIDCOL DEFAULT newid()
                         UNIQUE NOT NULL,                        -- Global unique record identifier
  PRIMARY KEY          (map_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE map_save

```
(
  map_partition        INTEGER NOT NULL,                         -- partition ID to which this map belongs
  map_public           SMALLINT,                                 -- 1 = public, 0 = private
  map_id               INTEGER NOT NULL,                         -- unique ID for this map
  map_name             CHAR(32) NOT NULL,                        -- map name
  map_type             SMALLINT,                                 -- 1 = top level map, 2 = normal map, 3, = popup map
  site                 CHAR(32) NOT NULL,                        -- Site Name
  map_guid             uniqueidentifier NOT NULL,                -- Global unique record identifier
  map_checksum         INTEGER NOT NULL,                         -- Record checksum
  upd_guid             uniqueidentifier NOT NULL,                -- Guid for update operation
  upd_mode             SMALLINT NOT NULL,                        -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp        DATETIME NOT NULL,                        -- Date / Time of the update
  upd_checksum         INTEGER NOT NULL,                         -- Checksum
  upd_rowguid          uniqueidentifier ROWGUIDCOL DEFAULT newid()
                         UNIQUE NOT NULL,                        -- Global unique record identifier
  PRIMARY KEY          ( map_partition, map_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE messagefilter

```
(
  mf_id                       INTEGER IDENTITY NOT FOR REPLICATION
                                PRIMARY KEY,                                  -- unique ID of this message filter
  mf_name                     CHAR(32) NOT NULL,                              -- message filter name
  mf_partition                INTEGER REFERENCES partition(part_number)
                                NOT FOR REPLICATION,                          -- partition ID to which this message filter belongs
  mf_public                   SMALLINT NOT NULL,                              -- 1=message filter is public, 0=message filter is private
  mf_fil_priority             TEXT NULL,                                      -- priority range filter
  mf_fil_sitename             TEXT NULL,                                      -- site name filter
  mf_fil_partition            TEXT NULL,                                      -- partition filter
  mf_fil_qstring              TEXT NULL,                                      -- query string filter
  mf_fil_operator             TEXT NULL,                                      -- operator filter
  mf_fil_msgtype              TEXT NULL,                                      -- message type filter
  mf_fil_itemname             TEXT NULL,                                      -- item name of message filter
  mf_fil_priority_type        SMALLINT NOT NULL,                              -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_sitename_type        SMALLINT NOT NULL,                              -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_partition_type       SMALLINT NOT NULL,                              -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_qstring_type         SMALLINT NOT NULL,                              -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_operator_type        SMALLINT NOT NULL,                              -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_msgtype_type         SMALLINT NOT NULL,                              -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_itemname_type        SMALLINT NOT NULL,                              -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_priority_empty       SMALLINT NOT NULL,                              -- 1=messages without priority pass filter criteria
  mf_fil_partition_empty      SMALLINT NOT NULL,                              -- 1=messages without partition pass filter criteria
  mf_fil_qstring_empty        SMALLINT NOT NULL,                              -- 1=messages without query string pass filter criteria
  mf_fil_itemname_empty       SMALLINT NOT NULL,                              -- 1=messages without item name pass filter criteria
  mf_fil_operator_empty       SMALLINT NOT NULL,                              -- 1=messages without operator pass filter criteria
  site                        CHAR(32) NOT NULL,                              -- Site Name
  mf_guid                     uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
  mf_checksum                 INTEGER NOT NULL,                               -- Record checksum
  mf_fil_alarmcategory        TEXT NULL,                                      --Alarm category filter
  mf_fil_alarmcategory_type   SMALLINT NOT NULL,                              -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_alarmcategory_empty  SMALLINT NOT NULL,                              -- 1=messages without alarmcategory pass filter criteria
  mf_fil_alarmescalation      TEXT NULL,                                      -- Alarm Escalation range filter
  mf_fil_alarmescalation_type SMALLINT NOT NULL,                              -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_alarmescalation_empty SMALLINT NOT NULL,                             -- 1=messages without priority pass filter criteria
);
  CREATE UNIQUE INDEX         mf_site_name ON messagefilter (site, mf_name);
```

CREATE TABLE messagefilter_save

```
(
  mf_id                       INTEGER NOT NULL,                               -- unique ID of this message filter
  mf_name                     CHAR(32) NOT NULL,                              -- message filter name
  mf_partition                INTEGER,                                        -- partition ID to which this message filter belongs
  mf_public                   SMALLINT NOT NULL,                              -- 1=message filter is public, 0=message filter is private
  mf_fil_priority             TEXT NULL,                                      -- priority range filter
  mf_fil_sitename             TEXT NULL,                                      -- site name filter
  mf_fil_partition            TEXT NULL,                                      -- partition filter
  mf_fil_qstring              TEXT NULL,                                      -- query string filter
  mf_fil_operator             TEXT NULL,                                      -- operator filter
  mf_fil_msgtype              TEXT NULL,                                      -- message type filter
  mf_fil_itemname             TEXT NULL,                                      -- item name of message filter
  mf_fil_priority_type        SMALLINT NOT NULL,                              -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_sitename_type        SMALLINT NOT NULL,                              -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_partition_type       SMALLINT NOT NULL,                              -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_qstring_type         SMALLINT NOT NULL,                              -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_operator_type        SMALLINT NOT NULL,                              -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_msgtype_type         SMALLINT NOT NULL,                              -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_itemname_type        SMALLINT NOT NULL,                              -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_priority_empty       SMALLINT NOT NULL,                              -- 1=messages without priority pass filter criteria
  mf_fil_partition_empty      SMALLINT NOT NULL,                              -- 1=messages without partition pass filter criteria
  mf_fil_qstring_empty        SMALLINT NOT NULL,                              -- 1=messages without query string pass filter criteria
  mf_fil_itemname_empty       SMALLINT NOT NULL,                              -- 1=messages without item name pass filter criteria
```

```
  mf_fil_operator_empty       SMALLINT NOT NULL,                          -- 1=messages without operator pass filter criteria
  site                        CHAR(32) NOT NULL,                          -- Site Name
  mf_guid                     uniqueidentifier NOT NULL,                  -- Global unique record identifier
  mf_checksum                 INTEGER NOT NULL,                           -- Record checksum
  upd_guid                    uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode                    SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp               DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum                INTEGER NOT NULL,                           -- Checksum
  upd_rowguid                 uniqueidentifier ROWGUIDCOL DEFAULT newid()
                                 UNIQUE NOT NULL,                         -- Global unique record identifier
  mf_fil_alarmcategory        TEXT NULL,                                  --Alarm category filter
  mf_fil_alarmcategory_type   SMALLINT NOT NULL,                          -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_alarmcategory_empty  SMALLINT NOT NULL,                          -- 1=messages without alarmcategory pass filter criteria
  mf_fil_alarmescalation      TEXT NULL,                                  -- Alarm Escalation range filter
  mf_fil_alarmescalation_type SMALLINT NOT NULL,                          -- 1=this criteria is include, 0=this criteria is exclude
  mf_fil_alarmescalation_empty SMALLINT NOT NULL,                         -- 1=messages without priority pass filter criteria
  PRIMARY KEY                 (mf_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE mfg_item

```
(
  mfgi_mfg_id          INTEGER REFERENCES msgfiltergroup(mfg_id)
                          NOT FOR REPLICATION,                            -- message filter group ID
  mfgi_mf_id           INTEGER REFERENCES messagefilter(mf_id)
                          NOT FOR REPLICATION,                            -- message filter ID
  site                 CHAR(32) NOT NULL,                                 -- Site Name
  mfgi_guid            uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  mfgi_checksum        INTEGER NOT NULL,                                  -- Record checksum
  mfgi_auto_added      SMALLINT NOT NULL,                                 --0 means manual added, 1 means auto added.
  PRIMARY KEY          ( mfgi_mfg_id, mfgi_mf_id)
);
```

CREATE TABLE mfg_item_save

```
(
  mfgi_mfg_id          INTEGER,                                           -- message filter group ID
  mfgi_mf_id           INTEGER,                                           -- message filter ID
  site                 CHAR(32) NOT NULL,                                 -- Site Name
  mfgi_guid            uniqueidentifier NOT NULL,                         -- Global unique record identifier
  mfgi_checksum        INTEGER NOT NULL,                                  -- Record checksum
  upd_guid             uniqueidentifier NOT NULL,                         -- Guid for update operation
  upd_mode             SMALLINT NOT NULL,                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp        DATETIME NOT NULL,                                 -- Date / Time of the update
  upd_checksum         INTEGER NOT NULL,                                  -- Checksum
  upd_rowguid          uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                -- Global unique record identifier
  mfgi_auto_added      SMALLINT NOT NULL,                                 --0 means manual added, 1 means auto added.
  PRIMARY KEY          ( mfgi_mfg_id, mfgi_mf_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE mifareencode

```
(
  me_sector            INTEGER NOT NULL,                                  -- the sector number for the data address
  me_block             INTEGER NOT NULL,                                  -- the block number within the sector for the data  address
  me_offset            INTEGER NOT NULL,                                  -- the offset number within the block for the data address
  me_key               CHAR (12) NOT NULL,                                -- smart card access key in Hex string
  me_key_type          SMALLINT NOT NULL,                                 -- access key type: A=0;B=1
  me_db_table_name     CHAR(32) NULL,                                     -- database table where this data comes from
  me_db_col_name       CHAR(64) NULL,                                     -- database column where this data comes from
  me_data_value        CHAR (16) NULL,                                    -- the data value to encode into a smartcard
  site                 CHAR(32) NOT NULL,                                 -- Site Name
  me_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
```

```
  me_checksum           INTEGER NOT NULL,                          -- Record checksum
  PRIMARY KEY           (me_guid)
)
```

CREATE TABLE mifareencode_save

```
(
  me_sector             INTEGER NOT NULL,                          -- the sector number for the data address
  me_block              INTEGER NOT NULL,                          -- the block number within the sector for the data  address
  me_offset             INTEGER NOT NULL,                          -- the offset number within the block for the data address
  me_key                CHAR (12) NOT NULL,                        -- smart card access key in Hex string
  me_key_type           SMALLINT NOT NULL,                         -- access key type: A=0;B=1
  me_db_table_name      CHAR(32) NULL,                             -- database table where this data comes from
  me_db_col_name        CHAR(64) NULL,                             -- database column where this data comes from
  me_data_value         CHAR (16) NULL,                            -- the data value to encode into a smartcard
  site                  CHAR(32) NOT NULL,                         -- Site Name
  me_guid               uniqueidentifier NOT NULL,                 -- Global unique record identifier
  me_checksum           INTEGER NOT NULL,                          -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                 -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                         -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                          -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                         -- Global unique record identifier
  PRIMARY KEY           ( me_guid, upd_mode desc, upd_guid )
)
```

CREATE TABLE mseagraphic

```
(
  mg_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
  mg_name               CHAR(32) NOT NULL,                              -- short name of MSEA graphic
  mg_fqn                VARCHAR(512) NOT NULL,                          -- fully qualified name of MSEA graphic
  site                  CHAR(32) NOT NULL,                              -- Site Name
  mg_checksum           INTEGER NOT NULL,                               -- Record checksum
  PRIMARY KEY           ( mg_name, site )
);
```

CREATE TABLE mseagraphic_save

```
(
  mg_guid               uniqueidentifier NOT NULL,                 -- Global unique record identifier
  mg_name               CHAR(32) NOT NULL,                         -- short name of MSEA graphic
  mg_fqn                VARCHAR(512) NOT NULL,                     -- fully qualified name of MSEA graphic
  site                  CHAR(32) NOT NULL,                         -- Site Name
  mg_checksum           INTEGER NOT NULL,                          -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                 -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                         -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                          -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                         -- Global unique record identifier
  PRIMARY KEY           ( mg_guid, upd_mode desc, upd_guid )
);
```

CREATE TABLE msgfiltergroup

```
(
  mfg_id                INTEGER IDENTITY NOT FOR REPLICATION
                          PRIMARY KEY,                             -- unique ID for message filter group
  mfg_name              CHAR(32) NOT NULL,                         -- message filter group name
  mfg_partition         INTEGER REFERENCES partition(part_number)
                          NOT FOR REPLICATION,                     -- partition ID to which this message filter group belongs
  mfg_public            SMALLINT NOT NULL,                         -- 1 = message filter group is public, 0 = message filter
                                                                      group is private
```

```
  site                  CHAR(32) NOT NULL,                              -- Site Name
  mfg_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
  mfg_checksum          INTEGER NOT NULL,                               -- Record checksum
);
  CREATE UNIQUE INDEX     mfg_site_name ON msgfiltergroup (site, mfg_name);
```

CREATE TABLE msgfiltergroup_save

```
(
  mfg_id                INTEGER NOT NULL,                   -- unique ID for message filter group
  mfg_name              CHAR(32) NOT NULL,                  -- message filter group name
  mfg_partition         INTEGER,                            -- partition ID to which this message filter group belongs
  mfg_public            SMALLINT NOT NULL,                  -- 1 = message filter group is public, 0 = message filter group is private
  site                  CHAR(32) NOT NULL,                  -- Site Name
  mfg_guid              uniqueidentifier NOT NULL,          -- Global unique record identifier
  mfg_checksum          INTEGER NOT NULL,                   -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,          -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                  -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                   -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                  -- Global unique record identifier
  PRIMARY KEY           (mfg_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE [dbo].MusterDefinition

```
(
Table defines each muster zone
  m_id                  INTEGER IDENTITY NOT FOR REPLICATION
                          UNIQUE,                           -- Unique identifier for each muster zone
  m_name                CHAR(32) NOT NULL,                  -- Muster zone name
  m_partition           INTEGER NOT NULL REFERENCES partition(part_number)
                          NOT FOR REPLICATION,              -- Partition of muster zone
  m_enabled             SMALLINT NOT NULL,                  -- 0 = disabled, 1 = enabled
  m_public              SMALLINT NOT NULL,                  -- 0 = private, 1 = public muster zone
  m_state               SMALLINT NOT NULL,                  -- If enabled, 0 = ready, 1 = running, 2 = stopped,
                                                              3 = aborted (database problem during a muster)
                                                              -1 = degraded operation (partial hardware down or disabled)
                                                              -2 = inoperable (all defining or muster readers down or disabled)
                                                              -3 = disabled (me_enabled will be 0)
  m_auto_control        INTEGER NULL REFERENCES station(stn_id)
                          NOT FOR REPLICATION,              -- NULL = muster control app not started on muster start
                                                            -- not NULL = id of workstation on which to start the
                                                            -- muster control application
  m_no_auto_report      SMALLINT NOT NULL,                  -- 0 = allow auto printing during muster
                                                            -- 1 = suppress auto printing
  m_report_interval     SMALLINT NOT NULL,                  -- Printing interval in minutes for running muster
  m_report_format       SMALLINT NOT NULL,                  -- 0 = one line format, 1 = multi line
  m_startup_rule        SMALLINT NOT NULL,                  -- 0 - n-1, indicating the startup rule to use in
                                                            -- determining the initial list of personnel in
                                                            -- the zone at muster start (n is number of rules)
  m_startup_time        SMALLINT NOT NULL,                  -- Time interval associated with some of the rules,
                                                            -- meaning depends on the rule
  m_startup_time_units  SMALLINT NOT NULL,                  -- Units for m_startup_time, 0 = minutes, 1 = hours, 2 = days
  m_number_of_shifts    SMALLINT NOT NULL,                  -- If shift rules determine the startup condition,
                                                            -- contains number of shifts, 1, 2, 3, otherwise is zero
  m_shift_1             INTEGER NULL REFERENCES timezone(tz_id)
                          NOT FOR REPLICATION,              -- Timezone for 1st shift
  m_shift_2             INTEGER NULL REFERENCES timezone(tz_id)
                          NOT FOR REPLICATION,              -- Timezone for 2nd shift
  m_shift_3             INTEGER NULL REFERENCES timezone(tz_id)
                          NOT FOR REPLICATION,              -- Timezone for 3rd shift
  m_no_in_x_it          SMALLINT NOT NULL,                  -- Do not use entry/exit status for initial condition
```

```
                                                                -- even if entry/exit control is possible
  m_allow_expansion              SMALLINT NOT NULL,             -- 0 = do not allow expansion of running muster
                                                                -- 1 = allow expansion
  m_use_only_valid_badging_on_startup  SMALLINT NOT NULL,       -- 0 = use latest badging to determine who is in zone
                                                                -- whether valid or invalid
                                                                -- 1 = use valid badging only
  m_muster_any_reader            SMALLINT NOT NULL,             -- 0 = only designated muster readers
                                                                -- 1 = allow any non defining reader for muster
  m_track_movement               SMALLINT NOT NULL,             -- 0 = no movement tracking
                                                                -- 1 = allow movement tracking
  m_popup_on_start_alarm         SMALLINT NOT NULL,             -- 0 = do not pop up, 1 = pop up on alarm
                                                                -- 2 = pop up on secure, 3 = pop up on both
  m_popup_on_status_alarm        SMALLINT NOT NULL,             -- Same for the status alarm
  m_start_alarm_text_set_id      INTEGER NULL REFERENCES alrminst(ai_id)
     NOT FOR REPLICATION,                                       -- Start alarm set instruction text ID if not null
  m_start_alarm_text_sec_id      INTEGER NULL REFERENCES alrminst(ai_id)
     NOT FOR REPLICATION,                                       -- Start alarm secure instruction text
  m_status_alarm_text_set_id     INTEGER NULL REFERENCES alrminst(ai_id)
     NOT FOR REPLICATION,                                       -- Same for the status alarm
  m_status_alarm_text_sec_id     INTEGER NULL REFERENCES alrminst(ai_id)
     NOT FOR REPLICATION,                                       -- Etc.
  m_muster_start_time            DATETIME NOT NULL,             -- Time at which the last muster started
  m_muster_stop_time             DATETIME NOT NULL,             -- Time at which the last muster stopped
  m_enable_demuster              SMALLINT NOT NULL,             -- 1 = demuster function enabled for zone, 0 = not enabled
  m_auto_print_enable            SMALLINT NOT NULL,             -- 1 if auto print is enabled, is only valid while a muster
                                                                -- is running, can't be interpreted if not running
  m_hardware_status              SMALLINT NOT NULL,             -- Hardware status 1 = OK, 0 = degraded, <0 = inoperable
                                                                -- set by muster service -- m_state also reflects this condition
  m_hardware_status_alarm_time   DATETIME NOT NULL,             -- Time of last hardware status alarm going set, if any
                                                                -- set by muster service
  m_popup_on_abort_alarm         SMALLINT NOT NULL,             -- Like above but for an abort alarm
  m_popup_on_disabled_alarm      SMALLINT NOT NULL,             -- Like above but for an alarm if a muster is triggered for a
                                                                   disabled zone
  m_abort_alarm_text_set_id      INTEGER NULL REFERENCES alrminst(ai_id)
     NOT FOR REPLICATION,                                       -- Instruction text for abort alarm, note, no secure text
  m_abort_alarm_text_sec_id      INTEGER NULL REFERENCES alrminst(ai_id)
     NOT FOR REPLICATION,                                       -- Going secure text for abort alarm
  m_disabled_alarm_text_set_id   INTEGER NULL REFERENCES alrminst(ai_id)
     NOT FOR REPLICATION,                                       -- Going set instruction for disabled alarm
  m_post_proc                    SMALLINT NOT NULL,             -- Valid only when m_state = 2, state of muster post processing
                                                                -- 0 = de-muster/save data not done, 1 = de-muster done
                                                                -- 2 = save data done, 3 = both have been done
  site                           CHAR(32) NOT NULL,             -- Site Name
  m_guid                         uniqueidentifier ROWGUIDCOL
     UNIQUE NOT NULL,                                           -- Global unique record identifier
  m_checksum                     INTEGER NOT NULL,              -- Record checksum
  PRIMARY KEY                    (m_partition, m_id)
) ;
  CREATE UNIQUE INDEX            m_site_name ON MusterDefinition (site, m_name);
```

CREATE TABLE MusterDefinition_save

```
(
Table defines each muster zone
  m_id                           INTEGER NOT NULL,              -- Unique identifier for each muster zone
  m_name                         CHAR(32) NOT NULL,             -- Muster zone name
  m_partition                    INTEGER NOT NULL,              -- Partition of muster zone
  m_enabled                      SMALLINT NOT NULL,             -- 0 = disabled, 1 = enabled
  m_public                       SMALLINT NOT NULL,             -- 0 = private, 1 = public muster zone
  m_state                        SMALLINT NOT NULL,             -- If enabled, 0 = ready, 1 = running, 2 = stopped,
                                                                -- 3 = aborted (database problem during a muster)
                                                                  -1 = degraded operation (partial hardware down or disabled)
                                                                  -2 = inoperable (all defining or muster readers down or disabled)
                                                                  - 3 = disabled (me_enabled will be 0)
```

```
    m_auto_control                  INTEGER NULL,                   -- NULL = muster control app not started on muster start
                                                                    -- not NULL = id of workstation on which to start the
                                                                    -- muster control application
    m_no_auto_report                SMALLINT NOT NULL,              -- 0 = allow auto printing during muster
                                                                    -- 1 = suppress auto printing
    m_report_interval               SMALLINT NOT NULL,              -- Printing interval in minutes for running muster
    m_report_format                 SMALLINT NOT NULL,              -- 0 = one line format, 1 = multi line
    m_startup_rule                  SMALLINT NOT NULL,              -- 0 - n-1, indicating the startup rule to use in
                                                                    -- determining the initial list of personnel in
                                                                    -- the zone at muster start (n is number of rules)
    m_startup_time                  SMALLINT NOT NULL,              -- Time interval associated with some of the rules,
                                                                    -- meaning depends on the rule
    m_startup_time_units            SMALLINT NOT NULL,              -- Units for m_startup_time, 0 = minutes, 1 = hours, 2 = days
    m_number_of_shifts              SMALLINT NOT NULL,              -- If shift rules determine the startup condition,
                                                                    -- contains number of shifts, 1, 2, 3, otherwise os zero
    m_shift_1                       INTEGER NULL,                   -- Timezone for 1st shift
    m_shift_2                       INTEGER NULL,                   -- Timezone for 2nd shift
    m_shift_3                       INTEGER NULL,                   -- Timezone for 3rd shift
    m_no_in_x_it                    SMALLINT NOT NULL,              -- Do not use entry/exit status for initial condition
                                                                    -- even if entry/exit control is possible
    m_allow_expansion               SMALLINT NOT NULL,              -- 0 = do not allow expansion of running muster
                                                                    -- 1 = allow expansion
    m_use_only_valid_badging_on_startup  SMALLINT NOT NULL,         -- 0 = use latest badging to determine who is in zone
                                                                    -- whether valid or invalid
                                                                    -- 1 = use valid badging only
    m_muster_any_reader             SMALLINT NOT NULL,              -- 0 = only designated muster readers
                                                                    -- 1 = allow any non defining reader for muster
    m_track_movement                SMALLINT NOT NULL,              -- 0 = no movement tracking
                                                                    -- 1 = allow movement tracking
    m_popup_on_start_alarm          SMALLINT NOT NULL,              -- 0 = do not pop up, 1 = pop up on alarm
                                                                    -- 2 = pop up on secure, 3 = pop up on both
    m_popup_on_status_alarm         SMALLINT NOT NULL,              -- Same for the status alarm
    m_start_alarm_text_set_id       INTEGER NULL,                   -- Start alarm set instruction text ID if not null
    m_start_alarm_text_sec_id       INTEGER NULL,                   -- Start alarm secure instruction text
    m_status_alarm_text_set_id      INTEGER NULL,                   -- Same for the status alarm
    m_status_alarm_text_sec_id      INTEGER NULL,                   -- Etc.
    m_muster_start_time             DATETIME NOT NULL,              -- Time at which the last muster started
    m_muster_stop_time              DATETIME NOT NULL,              -- Time at which the last muster stopped
    m_enable_demuster               SMALLINT NOT NULL,              -- 1 = demuster function enabled for zone, 0 = not enabled
    m_auto_print_enable             SMALLINT NOT NULL,              -- 1 if auto print is enabled, is only valid while a muster
                                                                    -- is running, can't be interpreted if not running
    m_hardware_status               SMALLINT NOT NULL,              -- Hardware status 1 = OK, 0 = degraded, <0 = inoperable
                                                                    -- set by muster service, -- m_state also reflects this condition
    m_hardware_status_alarm_time    DATETIME NOT NULL,              -- Time of last hardware status alarm going set, if any
                                                                    -- set by muster service
    m_popup_on_abort_alarm          SMALLINT NOT NULL,              -- Like above but for an abort alarm
    m_popup_on_disabled_alarm       SMALLINT NOT NULL,              -- Like above but for an alarm if a muster is triggered for a
                                                                      disabled zone
    m_abort_alarm_text_set_id       INTEGER NULL,                   -- Instruction text for abort alarm, note, no secure text
    m_abort_alarm_text_sec_id       INTEGER NULL,                   -- Going secure text for abort alarm
    m_disabled_alarm_text_set_id    INTEGER NULL,                   -- Going set instruction for disabled alarm
    m_post_proc                     SMALLINT NOT NULL,              -- Valid only when m_state = 2, state of muster post processing
                                                                    -- 0 = de-muster/save data not done, 1 = de-muster done
                                                                    -- 2 = save data done, 3 = both have been done
    site                            CHAR(32) NOT NULL,              -- Site Name
    m_guid                          uniqueidentifier NOT NULL,      -- Global unique record identifier
    m_checksum                      INTEGER NOT NULL,               -- Record checksum
    upd_guid                        uniqueidentifier NOT NULL,      -- Guid for update operation
    upd_mode                        SMALLINT NOT NULL,              -- Update mode (0 - delete, 1 - edit, 2 - new)
    upd_timestamp                   DATETIME NOT NULL,              -- Date / Time of the update
    upd_checksum                    INTEGER NOT NULL,               -- Checksum
    upd_rowguid                     uniqueidentifier ROWGUIDCOL DEFAULT
                                        newid() UNIQUE NOT NULL,    -- Global unique record identifier
    PRIMARY KEY                     (m_partition, m_id, upd_mode desc, upd_guid)
) ;
```

CREATE TABLE [dbo].MusterPrinters

```
(
Table defines the printers associated with muster zones
  mp_name              CHAR(255) NOT NULL,                                  -- Printer name
  mp_zone              INTEGER NOT NULL REFERENCES MusterDefinition(m_id)
                          NOT FOR REPLICATION,                              -- ID of Muster Zone using this printer
  site                 CHAR(32) NOT NULL,                                   -- Site Name
  mp_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,         -- Global unique record identifier
  mp_checksum          INTEGER NOT NULL,                                    -- Record checksum
  PRIMARY KEY          (mp_zone, mp_name),
) ;
```

CREATE TABLE MusterPrinters_save

```
(
  -- Table defines the printers associated with muster zones
  mp_name              CHAR(255) NOT NULL,                                  -- Printer name
  mp_zone              INTEGER NOT NULL,                                    -- ID of Muster Zone using this printer
  site                 CHAR(32) NOT NULL,                                   -- Site Name
  mp_guid              uniqueidentifier NOT NULL,                           -- Global unique record identifier
  mp_checksum          INTEGER NOT NULL,                                    -- Record checksum
  upd_guid             uniqueidentifier NOT NULL,                           -- Guid for update operation
  upd_mode             SMALLINT NOT NULL,                                   -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp        DATETIME NOT NULL,                                   -- Date / Time of the update
  upd_checksum         INTEGER NOT NULL,                                    -- Checksum
  upd_rowguid          uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                  -- Global unique record identifier
  PRIMARY KEY          (mp_zone, mp_name, upd_mode desc, upd_guid),
) ;
```

CREATE TABLE [dbo].MusterReaders

```
(
Table defines the readers for a muster zone
  mr_id                INTEGER IDENTITY NOT FOR REPLICATION
                          PRIMARY KEY,                                      -- Needed to provide a unique ky for this table
  mr_zone              INTEGER NOT NULL REFERENCES MusterDefinition(m_id)
                          NOT FOR REPLICATION,                              -- Zone to which this reader belongs
  mr_reader            INTEGER NULL REFERENCES terminal(tp_term_id)
                          NOT FOR REPLICATION,                              -- If not NULL, ID of reader associated with zone
  mr_reader_group      INTEGER NULL REFERENCES termgrp(tg_id)
                          NOT FOR REPLICATION,                              -- If not NULL, ID of terminal group associated with zone
  mr_type              INTEGER NOT NULL,                                    -- 0 = zone defining reader/group, 1 = muster reader/group,
                                                                            -- 2 = sequester reader/group
  mr_expansion_reader  SMALLINT NOT NULL,                                   -- 0 = permanent reader, 1 = temporary expansion reader
  site                 CHAR(32) NOT NULL,                                   -- Site Name
  mr_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,         -- Global unique record identifier
  mr_checksum          INTEGER NOT NULL,                                    -- Record checksum
) ;
  CREATE INDEX         mr_zone_readers ON MusterReaders(mr_zone, mr_type) ;
```

CREATE TABLE MusterReaders_save

```
(
Table defines the readers for a muster zone
  mr_id                INTEGER NOT NULL,                                    -- Needed to provide a unique ky for this table
  mr_zone              INTEGER NOT NULL,                                    -- Zone to which this reader belongs
  mr_reader            INTEGER NULL,                                        -- If not NULL, ID of reader associated with zone
  mr_reader_group      INTEGER NULL,                                        -- If not NULL, ID of terminal group associated with zone
  mr_type              INTEGER NOT NULL,                                    -- 0 = zone defining reader/group,
                                                                            -- 1 = muster reader/group, 2 = sequester reader/group
  mr_expansion_reader  SMALLINT NOT NULL,                                   -- 0 = permanent reader, 1 = temporary expansion reader
  site                 CHAR(32) NOT NULL,                                   -- Site Name
  mr_guid              uniqueidentifier NOT NULL,                           -- Global unique record identifier
```

```
  mr_checksum          INTEGER NOT NULL,                                  -- Record checksum
  upd_guid             uniqueidentifier NOT NULL,                         -- Guid for update operation
  upd_mode             SMALLINT NOT NULL,                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp        DATETIME NOT NULL,                                 -- Date / Time of the update
  upd_checksum         INTEGER NOT NULL,                                  -- Checksum
  upd_rowguid          uniqueidentifier ROWGUIDCOL DEFAULT newid()
                         UNIQUE NOT NULL,                                 -- Global unique record identifier
  PRIMARY KEY          (mr_id, upd_mode desc, upd_guid)
) ;
```

CREATE TABLE [dbo].MusterReportControl

```
(
  mrc_id               INTEGER NOT NULL IDENTITY NOT FOR
                         REPLICATION UNIQUE,                              -- Unique ID for this muster
  mrc_zone_name        CHAR(32) NOT NULL,                                 -- Unique Muster Zone ID
  mrc_zone_partition   CHAR(32) NOT NULL,                                 -- Partition name
  mrc_start_time       DATETIME NOT NULL,                                 -- Start time of muster
  mrc_stop_time        DATETIME NOT NULL,                                 -- Stop time of muster
  mrc_drill            SMALLINT NOT NULL,                                 -- 0 = real muster, 1 = drill only
  mrc_started_by       SMALLINT NOT NULL,                                 -- 0 = started by event (all types)
                                                                          -- 1 = started by operator
  mrc_start_id         CHAR(64) NULL,                                     -- Operator or event name depending on mrc_started_by
  site                 CHAR(32) NOT NULL,                                 -- Site Name
  mrc_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  mrc_checksum         INTEGER NOT NULL,                                  -- Record checksum
  PRIMARY KEY          (mrc_id, mrc_zone_name),
) ;
  CREATE INDEX         mrc_key ON MusterReportControl(site, mrc_zone_name, mrc_start_time);
  CREATE INDEX         mrc_stop_time ON MusterReportControl ( mrc_stop_time );
```

CREATE TABLE [dbo].MusterReportData

```
(
This table lists personnel remaining in each of the 7 groups at the end of a muster
  mrd_muster_id        INTEGER NOT NULL REFERENCES MusterReportControl(mrc_id)
                         NOT FOR REPLICATION,                             -- Unique ID of the muster
  mrd_reference        INTEGER NOT NULL,                                  -- Unique key to MusterReportControl
  mrd_type             SMALLINT NOT NULL,                                 -- Group type, 0 = at risk, 1 = trapped, 2 = wandering,
                                                                          -- 3 = mustered, 4 = sequestered, 5 = rescuer
  mrd_name             CHAR(64) NOT NULL,                                 -- Badge holder name string
  mrd_badge_number     VARCHAR(100) NOT NULL,                             -- Badge number
  mrd_last_location    CHAR(32) NOT NULL,                                 -- Last badge location (reader name)
  mrd_company_name     CHAR(32) NOT NULL,                                 -- Company name
  mrd_org_name         CHAR(32) NOT NULL,                                 -- Org name
  mrd_time             DATETIME NOT NULL,                                 -- Time that this person entered this group
  mrd_priority         SMALLINT NOT NULL,                                 -- Last action priority
  site                 CHAR(32) NOT NULL,                                 -- Site Name
  mrd_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  mrd_checksum         INTEGER NOT NULL,                                  -- Record checksum
  PRIMARY KEY          (mrd_reference, mrd_type, mrd_priority, mrd_badge_number),
) ;
  CREATE UNIQUE INDEX    mrd_key ON MusterReportData(site, mrd_reference, mrd_type, mrd_priority, mrd_badge_number);
```

CREATE TABLE ogdata

```
(
  og_group_id          INTEGER REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION,   -- output group ID
  og_outputpt_id       INTEGER REFERENCES output(op_output_id) NOT FOR REPLICATION,     -- output point ID
  site                 CHAR(32) NOT NULL,                                               -- Site Name
  og_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                     -- Global unique record identifier
  og_checksum          INTEGER NOT NULL,                                                -- Record checksum
  PRIMARY KEY          ( og_group_id, og_outputpt_id )
);
```

CREATE TABLE ogdata_save

```
(
  og_group_id           INTEGER NOT NULL,                                          -- output group ID
  og_outputpt_id        INTEGER NOT NULL,                                          -- output point ID
  site                  CHAR(32) NOT NULL,                                         -- Site Name
  og_guid               uniqueidentifier NOT NULL,                                 -- Global unique record identifier
  og_checksum           INTEGER NOT NULL,                                          -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                                 -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                         -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                          -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY           ( og_group_id, og_outputpt_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE optionkey

```
(
  ok_option_key         CHAR(40) PRIMARY KEY                                       NOT NULL,
  ok_option_id          INT NULL
);
  CREATE INDEX          ok_option_id ON optionkey ( ok_option_id );
```

CREATE TABLE otis_compass_mode

```
(
  otis_comp_mode_id             INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,    -- unique ID
  otis_comp_mode_elev_id        INTEGER NOT NULL REFERENCES elevator (el_elevator_id)
                                    NOT FOR REPLICATION,                        -- elevator ID
  otis_comp_mode_device_ref     INT,                                            -- device reference
  otis_comp_mode_mode           SMALLINT,                                       -- current DEC mode (1 - 4)
  otis_comp_mode_allowed_floors BINARY(32),                                     -- 32 bytes = allowed floors
  site                          CHAR(32) NOT NULL,                              -- Site Name
  otis_comp_mode_guid           uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
  otis_comp_mode_checksum       INTEGER NOT NULL,                               -- Record checksum
  PRIMARY KEY                   (otis_comp_mode_id)
);
  CREATE INDEX                  otis_comp_mode_elev_id ON otis_compass_mode( otis_comp_mode_elev_id );
  CREATE INDEX                  otis_comp_mode_device_ref ON otis_compass_mode( otis_comp_mode_device_ref );
```

CREATE TABLE otis_compass_mode_save

```
(
  otis_comp_mode_id             INTEGER NOT NULL,                               -- unique ID
  otis_comp_mode_elev_id        INTEGER,                                        -- elevator ID
  otis_comp_mode_device_ref     INT,                                            -- device reference
  otis_comp_mode_mode           SMALLINT,                                       -- current DEC mode (1 - 4)
  otis_comp_mode_allowed_floors BINARY(32),                                     -- 32 bytes define allowed floors
  site                          CHAR(32) NOT NULL,                              -- Site Name
  otis_comp_mode_guid           uniqueidentifier NOT NULL,                      -- Global unique record identifier
  otis_comp_mode_checksum       INTEGER NOT NULL,                               -- Record checksum
  upd_guid                      uniqueidentifier NOT NULL,                      -- Guid for update operation
  upd_mode                      SMALLINT NOT NULL,                              -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                 DATETIME NOT NULL,                              -- Date / Time of the update
  upd_checksum                  INTEGER NOT NULL,                               -- Checksum
  upd_rowguid                   uniqueidentifier ROWGUIDCOL DEFAULT newid()
                                    UNIQUE NOT NULL,                            -- Global unique record identifier
  PRIMARY KEY                   (otis_comp_mode_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE otis_compass_status

```
(
  otis_comp_status_id           INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,   -- unique ID
  otis_comp_status_device_ref   INT,                                           -- device reference
  otis_comp_status_device_type  SMALLINT NOT NULL,                             -- 0 = unknown, 1 = DEC, 2 = DES, 3 = DER
  otis_comp_status_status       SMALLINT NOT NULL,                             -- 0 = unknown, 1 = on-line, 2 = off-line
  otis_comp_status_timestamp    DATETIME NOT NULL,                             -- time stamp
  site                          CHAR(32) NOT NULL,                             -- Site Name
  otis_comp_status_guid         uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
  otis_comp_status_checksum     INTEGER NOT NULL,                              -- Record checksum
  PRIMARY KEY                   (otis_comp_status_id)
);
  CREATE INDEX                  otis_comp_status_device_ref ON otis_compass_status( otis_comp_status_device_ref );
```

CREATE TABLE output

```
(
  op_partition          INTEGER REFERENCES partition(part_number)
                          NOT FOR REPLICATION,                  -- ID of partition for this input
  op_public             SMALLINT NOT NULL,                      -- publication
  op_output_id          INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,  -- Unique database ID of output point
  op_point_name         varchar(32) NOT NULL,                   -- name of output point
  op_enable             SMALLINT NOT NULL,                      -- point enabled or not
  op_point_number       SMALLINT NOT NULL,                      -- output number
  op_term_id            INTEGER REFERENCES terminal(tp_term_id)
                          NOT FOR REPLICATION,                  -- terminal ID that has this output
  op_query_string       CHAR(64),                               -- query string enter by user
  op_active_state       SMALLINT NOT NULL,                      -- active state (0-4)
  op_duration           SMALLINT NOT NULL,                      -- duration in secs (only for timed active state)
  site                  CHAR(32) NOT NULL,                      -- Site Name
  op_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  op_checksum           INTEGER NOT NULL,                       -- Record checksum
  PRIMARY KEY           ( op_partition, op_output_id)
);
  CREATE INDEX              op_public ON output(op_public);
  CREATE UNIQUE INDEX       op_term_id ON output ( op_term_id, op_point_number );
  CREATE UNIQUE INDEX       op_site_name ON output (site, op_point_name);
```

CREATE TABLE outputgrp

```
(
  op_group_partition    INTEGER REFERENCES partition(part_number)
                          NOT FOR REPLICATION,                  -- partition ID to which this group belongs
  op_group_public       SMALLINT,                               -- 1 = public, 0 = private
  op_group_id           INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,  -- unique ID for this group
  op_group_name         CHAR(32) NOT NULL,                      -- group name
  op_group_panel_id     INTEGER REFERENCES panel(p_panel_id)
                          NOT FOR REPLICATION,                  -- panel ID to which this group belongs
  op_group_index        SMALLINT,                               -- group panel index (1-598)
  site                  CHAR(32) NOT NULL,                      -- Site Name
  op_group_guid         uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  op_group_checksum     INTEGER NOT NULL,                       -- Record checksum
  PRIMARY KEY           ( op_group_partition, op_group_id)
);
  CREATE INDEX              op_group_public ON outputgrp ( op_group_public );
  CREATE INDEX              op_group_panel_id ON outputgrp ( op_group_panel_id);
  CREATE INDEX              op_group_index ON outputgrp ( op_group_index);
  CREATE UNIQUE INDEX       op_site_name ON outputgrp (site, op_group_name);
```

CREATE TABLE outputgrp_save

```
(
  op_group_partition    INTEGER,                                -- partition ID to which this group belongs
  op_group_public       SMALLINT,                               -- 1 = public, 0 = private
```

```
  op_group_id           INTEGER NOT NULL,                                                  -- unique ID for this group
  op_group_name         CHAR(32) NOT NULL,                                                 -- group name
  op_group_panel_id     INTEGER,                                                           -- panel ID to which this group belongs
  op_group_index        SMALLINT,                                                          -- group panel index (1-598)
  site                  CHAR(32) NOT NULL,                                                 -- Site Name
  op_group_guid         uniqueidentifier NOT NULL,                                         -- Global unique record identifier
  op_group_checksum     INTEGER NOT NULL,                                                  -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                                         -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                                 -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                                  -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,       -- Global unique record identifier
  PRIMARY KEY           ( op_group_partition, op_group_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE output_status

```
(
  opstat_output_id      INTEGER REFERENCES output(op_output_id)
                          NOT FOR REPLICATION,                                             -- output ID to which this status belongs
  opstat_status         SMALLINT NOT NULL,                                                 -- last output status (0-2) (set by comms)
  opstat_timestamp      DATETIME NOT NULL,                                                 -- time of last output status (set by comms)
  site                  CHAR(32) NOT NULL,                                                 -- Site Name
  opstat_guid           uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                       -- Global unique record identifier
  opstat_checksum       INTEGER NOT NULL,                                                  -- Record checksum
  PRIMARY KEY           (opstat_output_id)
);
```

CREATE TABLE output_ck

```
(
  op_id                 INTEGER REFERENCES output(op_output_id)
                          NOT FOR REPLICATION,                                             -- Unique database ID of this output
  site                  CHAR(32) NOT NULL,                                                 -- Site Name
  op_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                       -- Global unique record identifier
  op_checksum           INTEGER NOT NULL,                                                  -- Record checksum
  PRIMARY KEY           (op_id)
);
```

CREATE TABLE output_ck_save

```
(
  op_id                 INTEGER NOT NULL,                                                  -- Unique database ID of this output
  site                  CHAR(32) NOT NULL,                                                 -- Site Name
  op_guid               uniqueidentifier NOT NULL,                                         -- Global unique record identifier
  op_checksum           INTEGER NOT NULL,                                                  -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                                         -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                                 -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                                  -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,       -- Global unique record identifier
  PRIMARY KEY           (op_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE output_p900

```
(
  opp_id                INTEGER REFERENCES output(op_output_id)
                          NOT FOR REPLICATION,                                             -- Unique database ID of this output
  opp_tz                INTEGER NULL REFERENCES timezone(tz_id)
                          NOT FOR REPLICATION,                                             -- timezone ID when output activated
  opp_action_term       INTEGER NULL REFERENCES terminal(tp_term_id)
                          NOT FOR REPLICATION,                                             -- terminal ID that has this output
  opp_action            SMALLINT NULL,                                                     -- output action
```

```
  opp_report            SMALLINT,                                   -- report priority -0: none, 1: alarm, 2: transact.
  opp_pp_value          SMALLINT,                                   -- to determine both how the output is activated, and the
                                                                       type of access which causes it to be activated
  site                  CHAR(32) NOT NULL,                          -- Site Name
  opp_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  opp_checksum          INTEGER NOT NULL,                           -- Record checksum
  PRIMARY KEY           (opp_id)
);
```

CREATE TABLE output_p900_save

```
(
  opp_id                INTEGER NOT NULL,                           -- Unique database ID of this output
  opp_tz                INTEGER NULL,                               -- timezone ID when output activated
  opp_action_term       INTEGER NULL,                               -- terminal ID that has this output
  opp_action            SMALLINT NULL,                              -- output action
  opp_report            SMALLINT,                                   -- report priority -0: none, 1: alarm, 2: transact.
  opp_pp_value          SMALLINT,                                   -- to determine both how the output is activated, and the
                                                                       type of access which causes it to be activated
  site                  CHAR(32) NOT NULL,                          -- Site Name
  opp_guid              uniqueidentifier NOT NULL,                  -- Global unique record identifier
  opp_checksum          INTEGER NOT NULL,                           -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                           -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                          -- Global unique record identifier
  PRIMARY KEY           (opp_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE output_s321ip

```
(
  opps321ip_output_id           INTEGER REFERENCES output(op_output_id)
                                  NOT FOR REPLICATION,              -- ID of output
  opps321ip_output_flash        SMALLINT NOT NULL,                  -- Flash rate 0 - 59 seconds
  opps321ip_enable_status_msg   SMALLINT NOT NULL,                  -- 1 = enable output status messages
  site                          CHAR(32) NOT NULL,                  -- Site Name
  opps321ip_guid                uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  opps321ip_checksum            INTEGER NOT NULL,                   -- Record checksum
  PRIMARY KEY                   ( opps321ip_output_id )
);
```

CREATE TABLE output_s321ip_save

```
(
  opps321ip_output_id           INTEGER NOT NULL,                   -- ID of output
  opps321ip_output_flash        SMALLINT NOT NULL,                  -- Flash rate 0 - 59 seconds
  opps321ip_enable_status_msg   SMALLINT NOT NULL,                  -- 1 = enable output status messages
  site                          CHAR(32) NOT NULL,                  -- Site Name
  opps321ip_guid                uniqueidentifier NOT NULL,          -- Global unique record identifier
  opps321ip_checksum            INTEGER NOT NULL,                   -- Record checksum
  upd_guid                      uniqueidentifier NOT NULL,          -- Guid for update operation
  upd_mode                      SMALLINT NOT NULL,                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                 DATETIME NOT NULL,                  -- Date / Time of the update
  upd_checksum                  INTEGER NOT NULL,                   -- Checksum
  upd_rowguid                   uniqueidentifier ROWGUIDCOL DEFAULT newid()
                                  UNIQUE NOT NULL,                  -- Global unique record identifier
  PRIMARY KEY                   ( opps321ip_output_id, upd_mode desc, upd_guid  )
);
```

CREATE TABLE output_save

```
(
  op_partition          INTEGER NOT NULL,                                  -- ID of partition for this input
  op_public             SMALLINT NOT NULL,                                 -- publication
  op_output_id          INTEGER NOT NULL,                                  -- Unique database ID of output point
  op_point_name         varchar(32) NOT NULL,                              -- name of output point
  op_enable             SMALLINT NOT NULL,                                 -- point enabled or not
  op_point_number       SMALLINT NOT NULL,                                 -- output number
  op_term_id            INTEGER,                                           -- terminal ID that has this output
  op_query_string       CHAR(64),                                          -- query string enter by user
  op_active_state       SMALLINT NOT NULL,                                 -- active state (0-4)
  op_duration           SMALLINT NOT NULL,                                 -- duration in secs (only for timed active state)
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  op_guid               uniqueidentifier NOT NULL,                         -- Global unique record identifier
  op_checksum           INTEGER NOT NULL,                                  -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                         -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                 -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                  -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                 -- Global unique record identifier
  PRIMARY KEY           ( op_partition, op_output_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE p2kadxmap

```
(
  am_SiteDirectorName   VARCHAR(50) NOT NULL,                              -- Site director's name
  am_SiteDirectorIP     VARCHAR(50) NOT NULL,                              -- Site director's ip address
  am_DeviceID           VARCHAR(50) NOT NULL,                              -- registration device id
  am_ADSRepository      VARCHAR(50) NOT NULL,                              -- ADS Reposistory name
  am_HostIP             VARCHAR(50) NOT NULL,                              -- P2000 Host IP address
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  am_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  am_checksum           INT NOT NULL,
  PRIMARY KEY           ( am_SiteDirectorName, site )
);
```

CREATE TABLE p2kadxmap_save

```
(
  am_SiteDirectorName   VARCHAR(50) NOT NULL,                              -- Site director's name
  am_SiteDirectorIP     VARCHAR(50) NOT NULL,                              -- Site director's ip address
  am_DeviceID           VARCHAR(50) NOT NULL,                              -- registration device id
  am_ADSRepository      VARCHAR(50) NOT NULL,                              -- ADS Reposistory name
  am_HostIP             VARCHAR(50) NOT NULL,                              -- P2000 Host IP address
  site                  CHAR(32) NOT NULL,                                 -- Site Name
  am_guid               uniqueidentifier NOT NULL,                         -- Global unique record identifier
  am_checksum           INT NOT NULL,
  upd_guid              uniqueidentifier NOT NULL,                         -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                 -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                  -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,  -- Global unique record identifier
  PRIMARY KEY           ( am_SiteDirectorName, site, upd_mode desc, upd_guid )
);
```

CREATE TABLE P900sysparameters

```
(
  sp_facility_code      SMALLINT NOT NULL,                                 -- 0 = disable, 1 = alarm, 2 = report
  sp_access_group       SMALLINT NOT NULL,                                 -- 0 = disable, 1 = alarm, 2 = report
  sp_disabled_badge     SMALLINT NOT NULL,                                 -- 0 = disable, 1 = alarm, 2 = report
  sp_issue_level        SMALLINT NOT NULL,                                 -- 0 = disable, 1 = alarm, 2 = report
```

```
    sp_badge_not_found          SMALLINT NOT NULL,                  -- 0 = disable, 1 = alarm, 2 = report
    sp_timezone                 SMALLINT NOT NULL,                  -- 0 = disable, 1 = alarm, 2 = report
    sp_pin_violation            SMALLINT NOT NULL,                  -- 0 = disable, 1 = alarm, 2 = report
    sp_local_anti_passback      SMALLINT NOT NULL,                  -- 0 = disable, 1 = alarm, 2 = report
    sp_valid_not_open           SMALLINT NOT NULL,                  -- 0 = disable, 1 = alarm, 2 = report
    sp_facility_code_p          SMALLINT NOT NULL,                  -- 0 = disable, 1 = alarm, 2 = report
    sp_access_group_p           SMALLINT NOT NULL,                  -- 1 = popup
    sp_disabled_badge_p         SMALLINT NOT NULL,                  -- 1 = popup
    sp_issue_level_p            SMALLINT NOT NULL,                  -- 1 = popup
    sp_badge_not_found_p        SMALLINT NOT NULL,                  -- 1 = popup
    sp_timezone_p               SMALLINT NOT NULL,                  -- 1 = popup
    sp_pin_violation_p          SMALLINT NOT NULL,                  -- 1 = popup
    sp_local_anti_passback_p    SMALLINT NOT NULL,                  -- 1 = popup
    sp_valid_not_open_p         SMALLINT NOT NULL,                  -- 1 = popup
    sp_facility_code_i          INTEGER NULL REFERENCES alrminst(ai_id)
                                    NOT FOR REPLICATION,            -- facility code alarm instruction text ID if not null
    sp_access_group_i           INTEGER NULL REFERENCES alrminst(ai_id)
                                    NOT FOR REPLICATION,            -- access group alarm instruction text ID if not null
    sp_disabled_badge_i         INTEGER NULL REFERENCES alrminst(ai_id)
                                    NOT FOR REPLICATION,            -- disabled badge alarm instruction text ID if not null
    sp_issue_level_i            INTEGER NULL REFERENCES alrminst(ai_id)
                                    NOT FOR REPLICATION,            -- issue level alarm instruction text ID if not null
    sp_badge_not_found_i        INTEGER NULL REFERENCES alrminst(ai_id)
                                    NOT FOR REPLICATION,            -- badge not found alarm instruction text ID if not null
    sp_timezone_i               INTEGER NULL REFERENCES alrminst(ai_id)
                                    NOT FOR REPLICATION,            -- timezone alarm instruction text ID if not null
    sp_pin_violation_i          INTEGER NULL REFERENCES alrminst(ai_id)
                                    NOT FOR REPLICATION,            -- pin violation alarm instruction text ID if not null
    sp_local_anti_passback_i    INTEGER NULL REFERENCES alrminst(ai_id)
                                    NOT FOR REPLICATION,            -- local anti-passback alarm instruction text ID if not null
    sp_valid_not_open_i         INTEGER NULL REFERENCES alrminst(ai_id)
                                    NOT FOR REPLICATION,            -- local anti-passback alarm instruction text ID if not null
    sp_holdoff_time             SMALLINT NOT NULL,                  -- holdoff time, 0 - as door relay time, otherwise 1-255 seconds
    sp_repeat_xaction_time      SMALLINT NOT NULL,                  -- 0 - 255 seconds
    sp_local_forgiveness        SMALLINT NOT NULL,                  -- 0 - disable, 1 - enabled, 2 - reset(local forgiveness NOW)
    sp_local_forgiveness_time   DATETIME NOT NULL,
    sp_card_mode                SMALLINT NOT NULL,
    site                        CHAR(32) NOT NULL PRIMARY KEY,      -- Site Name
    sp_guid                     uniqueidentifier ROWGUIDCOL
                                    UNIQUE NOT NULL,                -- Global unique record identifier
    sp_checksum                 INTEGER NOT NULL,                   -- Record checksum
);
```

CREATE TABLE P900sysparameters_save

```
(
    sp_facility_code            SMALLINT NOT NULL,                  -- 0 = disable, 1 = alarm, 2 = report
    sp_access_group             SMALLINT NOT NULL,                  -- 0 = disable, 1 = alarm, 2 = report
    sp_disabled_badge           SMALLINT NOT NULL,                  -- 0 = disable, 1 = alarm, 2 = report
    sp_issue_level              SMALLINT NOT NULL,                  -- 0 = disable, 1 = alarm, 2 = report
    sp_badge_not_found          SMALLINT NOT NULL,                  -- 0 = disable, 1 = alarm, 2 = report
    sp_timezone                 SMALLINT NOT NULL,                  -- 0 = disable, 1 = alarm, 2 = report
    sp_pin_violation            SMALLINT NOT NULL,                  -- 0 = disable, 1 = alarm, 2 = report
    sp_local_anti_passback      SMALLINT NOT NULL,                  -- 0 = disable, 1 = alarm, 2 = report
    sp_valid_not_open           SMALLINT NOT NULL,                  -- 0 = disable, 1 = alarm, 2 = report
    sp_facility_code_p          SMALLINT NOT NULL,                  -- 0 = disable, 1 = alarm, 2 = report
    sp_access_group_p           SMALLINT NOT NULL,                  -- 1 = popup
    sp_disabled_badge_p         SMALLINT NOT NULL,                  -- 1 = popup
    sp_issue_level_p            SMALLINT NOT NULL,                  -- 1 = popup
    sp_badge_not_found_p        SMALLINT NOT NULL,                  -- 1 = popup
    sp_timezone_p               SMALLINT NOT NULL,                  -- 1 = popup
    sp_pin_violation_p          SMALLINT NOT NULL,                  -- 1 = popup
    sp_local_anti_passback_p    SMALLINT NOT NULL,                  -- 1 = popup
    sp_valid_not_open_p         SMALLINT NOT NULL,                  -- 1 = popup
```

```
  sp_facility_code_i        INTEGER NULL,                              -- facility code alarm instruction text ID if not null
  sp_access_group_i         INTEGER NULL,                              -- access group alarm instruction text ID if not null
  sp_disabled_badge_i       INTEGER NULL,                              -- disabled badge alarm instruction text ID if not null
  sp_issue_level_i          INTEGER NULL,                              -- issue level alarm instruction text ID if not null
  sp_badge_not_found_i      INTEGER NULL,                              -- badge not found alarm instruction text ID if not null
  sp_timezone_i             INTEGER NULL,                              -- timezone alarm instruction text ID if not null
  sp_pin_violation_i        INTEGER NULL,                              -- pin violation alarm instruction text ID if not null
  sp_local_anti_passback_i  INTEGER NULL,                              -- local anti-passback alarm instruction text ID if not null
  sp_valid_not_open_i       INTEGER NULL,                              -- local anti-passback alarm instruction text ID if not null
  sp_holdoff_time           SMALLINT NOT NULL,                         -- holdoff time, 0 - as door relay time, otherwise 1-255 seconds
  sp_repeat_xaction_time    SMALLINT NOT NULL,                         -- 0 - 255 seconds
  sp_local_forgiveness      SMALLINT NOT NULL,                         -- 0 - disable, 1 - enabled, 2 - reset(local forgiveness NOW)
  sp_local_forgiveness_time DATETIME NOT NULL,
  sp_card_mode              SMALLINT NOT NULL,
  site                      CHAR(32) NOT NULL,                         -- Site Name
  sp_guid                   uniqueidentifier NOT NULL,                 -- Global unique record identifier
  sp_checksum               INTEGER NOT NULL,                          -- Record checksum
  upd_guid                  uniqueidentifier NOT NULL,                 -- Guid for update operation
  upd_mode                  SMALLINT NOT NULL,                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp             DATETIME NOT NULL,                         -- Date / Time of the update
  upd_checksum              INTEGER NOT NULL,                          -- Checksum
  upd_rowguid               uniqueidentifier ROWGUIDCOL DEFAULT newid()
                              UNIQUE NOT NULL,                         -- Global unique record identifier
  PRIMARY KEY               (site, upd_mode desc, upd_guid)
);
```

CREATE TABLE panel

```
(
  p_partition               INTEGER REFERENCES partition(part_number)
                              NOT FOR REPLICATION,                     -- partition ID to which this panel belongs
  p_public                  SMALLINT,                                  -- 1 = panel is public, 0 = panel is private
  p_unit_name               VARCHAR(32) NOT NULL,                      -- user entered name
  p_panel_id                INTEGER IDENTITY NOT FOR
                              REPLICATION UNIQUE,                      -- unique database ID
  p_query_string            CHAR(64) NULL,                             -- query string entered by user
  p_enabled                 SMALLINT NOT NULL,                         -- 1 = enabled
  p_code_major_version      SMALLINT,                                  -- Panel firmware revision major portion
  p_code_minor_version      SMALLINT,                                  -- Panel firmware revision minor portion
  p_code_build_version      SMALLINT,                                  -- Panel firmware revision build portion
  p_ipl_major_version       SMALLINT,                                  -- Panel ipl firmware revision major portion
  p_ipl_minor_version       SMALLINT,                                  -- Panel ipl firmware revision minor portion
  p_ipl_build_version       SMALLINT,                                  -- Panel ipl firmware revision build portion
  p_min_major_version       SMALLINT NOT NULL,                         -- Minimum version of panel firmware that can use this record
                                                                         major portion
  p_min_minor_version       SMALLINT NOT NULL,                         -- Minimum version of panel firmware that can use this record
                                                                         minor portion
  p_min_build_version       SMALLINT NOT NULL,                         -- Minimum version of panel firmware that can use this record
                                                                         build portion
  p_max_major_version       SMALLINT NOT NULL,                         -- Maximum version of panel firmware that can use this record
                                                                         major portion
  p_max_minor_version       SMALLINT NOT NULL,                         -- Maximum version of panel firmware that can use this record
                                                                         minor portion
  p_max_build_version       SMALLINT NOT NULL,                         -- Maximum version of panel firmware that can use this record
                                                                         build portion
  p_configuration_model     SMALLINT NOT NULL,                         -- Panel model identifier 0 = Unknown, 1 = CK720, 2 = CK705,
                                                                         3 = CK710
  p_hardware_model          SMALLINT NOT NULL,                         -- Panel model identifier 0 = Unknown, 1 = CK720, 2 = CK705,
                                                                         3 = CK710
  p_version_text            VARCHAR(70) NULL,                          -- panel version string (set by comms)
  p_time_date_offset        INTEGER NOT NULL,                          -- time offset in mins
  p_offset_enable           SMALLINT NOT NULL,                         -- 1 = enable time offset
  p_ip_pri                  CHAR(16) NULL,                             -- primary IP address
  p_ip_alt                  CHAR(16) NULL,                             -- alternate IP address
```

```
  p_pri_comm_pref              SMALLINT NOT NULL,                    -- primary comm is preferred path
  p_net_timeout                SMALLINT NOT NULL,                    -- network timeout in secs
  p_lan_net_polling_interval   INTEGER NULL,                         -- Downloaded to panel for primary channel polling
  p_dialup_net_polling_interval INTEGER NULL,                        -- Downloaded to panel for alternate channel polling
  p_lan_net_polling_timeout    INTEGER NULL,                         -- Host uses for primary channel polling
  p_dialup_net_polling_timeout INTEGER NULL,                         -- Host uses for alternate channel polling
  p_unit                       INTEGER NOT NULL,                     -- panel address
  p_poll_timeout               SMALLINT NOT NULL,                    -- Time in milliseconds to wait for a poll response
  p_poll_direction             SMALLINT NOT NULL,                    -- Preferred poll direction, 0 = forward, 1 = reverse
  p_actual_poll_dir            SMALLINT NOT NULL,                    -- Actual poll direction, 0 = forward, 1 = reverse, -1 = unknown
  p_retry_delay                SMALLINT NOT NULL,                    -- re establish delay for serial panels in seconds
  p_inp_report_delay           SMALLINT NOT NULL,                    -- input reporting delay
  p_out_report_delay           SMALLINT NOT NULL,                    -- output reporting delay
  p_tz_checking                SMALLINT NOT NULL,                    -- 1 = timezone checking
  p_pin_type                   SMALLINT NOT NULL,                    -- 0 = algorithmic, 1 = custom
  p_4_or_5_digit               SMALLINT NOT NULL,                    -- 0 = 4 digits, 1 = 5 digits, 2 = 6 digits, etc.
  p_inxit                      SMALLINT NOT NULL,                    -- 1 = entry/exit
  p_upload                     SMALLINT NOT NULL,                    -- 1 = upload history
  site                         CHAR(32) NOT NULL,                    -- Site Name
  p_guid                       uniqueidentifier ROWGUIDCOL
                                 UNIQUE NOT NULL,                    -- Global unique record identifier
  p_checksum                   INTEGER NOT NULL,                     -- Record checksum
  p_bacnet                     SMALLINT NOT NULL,                    -- 1 = bacnet useses this panel.
  p_loop_id                    INTEGER NULL REFERENCES loopconfig(lc_id)
                                 NOT FOR REPLICATION,                -- loop number
  p_encryption_enabled         SMALLINT NOT NULL,                    -- 1 = encryption enabled
  p_encryption_key             VARCHAR(256) NULL,                    -- encryption key up to 256 characters
  PRIMARY KEY                  ( p_partition, p_panel_id )
);
  CREATE INDEX                 p_public ON panel( p_public );
  CREATE INDEX                 p_ip_pri ON panel ( p_ip_pri );
  CREATE INDEX                 p_ip_alt ON panel ( p_ip_alt );
  CREATE UNIQUE INDEX          p_site_name ON panel (site, p_unit_name);
```

CREATE TABLE panelholiday

```
(
  ph_panel_id     INTEGER REFERENCES panel(p_panel_id) NOT FOR REPLICATION,   -- panel ID
  ph0             INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph1             INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph2             INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph3             INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph4             INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph5             INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph6             INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph7             INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph8             INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph9             INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph10            INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph11            INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph12            INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph13            INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph14            INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph15            INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph16            INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph17            INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph18            INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph19            INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph20            INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph21            INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph22            INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph23            INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph24            INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
  ph25            INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION, -- holiday ID (NULL if empty)
```

```
  ph26          INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION,   -- holiday ID (NULL if empty)
  ph27          INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION,   -- holiday ID (NULL if empty)
  ph28          INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION,   -- holiday ID (NULL if empty)
  ph29          INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION,   -- holiday ID (NULL if empty)
  ph30          INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION,   -- holiday ID (NULL if empty)
  ph31          INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION,   -- holiday ID (NULL if empty)
  ph32          INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION,   -- holiday ID (NULL if empty)
  ph33          INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION,   -- holiday ID (NULL if empty)
  ph34          INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION,   -- holiday ID (NULL if empty)
  ph35          INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION,   -- holiday ID (NULL if empty)
  ph36          INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION,   -- holiday ID (NULL if empty)
  ph37          INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION,   -- holiday ID (NULL if empty)
  ph38          INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION,   -- holiday ID (NULL if empty)
  ph39          INTEGER NULL REFERENCES holiday(ho_id) NOT FOR REPLICATION,   -- holiday ID (NULL if empty)
  site          CHAR(32) NOT NULL,                                           -- Site Name
  ph_guid       uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                 -- Global unique record identifier
  ph_checksum   INTEGER NOT NULL,                                            -- Record checksum
  PRIMARY KEY   (ph_panel_id)
);
```

CREATE TABLE panelholiday_save

```
(
  ph_panel_id   INTEGER NOT NULL,   -- panel ID
  ph0           INTEGER NULL,       -- holiday ID (NULL if empty)
  ph1           INTEGER NULL,       -- holiday ID (NULL if empty)
  ph2           INTEGER NULL,       -- holiday ID (NULL if empty)
  ph3           INTEGER NULL,       -- holiday ID (NULL if empty)
  ph4           INTEGER NULL,       -- holiday ID (NULL if empty)
  ph5           INTEGER NULL,       -- holiday ID (NULL if empty)
  ph6           INTEGER NULL,       -- holiday ID (NULL if empty)
  ph7           INTEGER NULL,       -- holiday ID (NULL if empty)
  ph8           INTEGER NULL,       -- holiday ID (NULL if empty)
  ph9           INTEGER NULL,       -- holiday ID (NULL if empty)
  ph10          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph11          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph12          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph13          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph14          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph15          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph16          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph17          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph18          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph19          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph20          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph21          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph22          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph23          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph24          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph25          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph26          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph27          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph28          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph29          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph30          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph31          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph32          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph33          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph34          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph35          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph36          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph37          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph38          INTEGER NULL,       -- holiday ID (NULL if empty)
  ph39          INTEGER NULL,       -- holiday ID (NULL if empty)
```

```
  site                  CHAR(32) NOT NULL,                                          -- Site Name
  ph_guid               uniqueidentifier NOT NULL,                                  -- Global unique record identifier
  ph_checksum           INTEGER NOT NULL,                                           -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                                  -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                          -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                           -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY           (ph_panel_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE panelstatus

```
(
  ps_panel_id           INTEGER REFERENCES panel(p_panel_id)
                          NOT FOR REPLICATION,          -- unique database ID
  ps_pref_comm_valid    SMALLINT,                       -- primary comm is valid (calculated)
  ps_status             SMALLINT,                       -- up/down status (set by comms)
  ps_status_timestamp   DATETIME,                       -- status timestamp (set by comms)
  ps_last_communication DATETIME NOT NULL,              -- last communication timestamp (set by comms)
  ps_dnld_delay_until   DATETIME NOT NULL,              -- Error Delay hold-off (Backoff Algorithm)
  ps_dnld_conn_fails    SMALLINT NOT NULL,              -- # of consuccutive failures spanning service sessions
  ps_dnld_conn_fails_total INTEGER NOT NULL,            -- Total # of consuccutive failures (cleared manually)
  ps_dnld_xfer_fails    SMALLINT NOT NULL,              -- # of consuccutive failures spanning service sessions
  ps_dnld_xfer_fails_total INTEGER NOT NULL,            -- Total # of consuccutive failures (cleared manually)
  ps_stats_cleared      DATETIME NOT NULL,              -- time stats were cleared (set manually)
  ps_clock_drift        FLOAT NOT NULL,                 -- avg. clock drift in the panel
  ps_max_clock_drift    INTEGER NOT NULL,               -- max clock drift seen from the panel
  ps_port_configuration SMALLINT NOT NULL,              -- 0 = CK720 V2.0 and earlier port usage
                                                        -- 1 = CK720 V2.1 port usage
                                                        -- 2 = CK720 V2.2 port usage
  site                  CHAR(32) NOT NULL,              -- Site Name
  ps_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  ps_checksum           INTEGER NOT NULL,               -- Record checksum
  ps_SN                 CHAR(22),                       -- Panel Serial Number if sent
  ps_details_timestamp  DATETIME NULL,
  ps_details1           VARCHAR(64) NULL,
  ps_details2           VARCHAR(64) NULL,
  PRIMARY KEY           (ps_panel_id)
);
```

CREATE TABLE paneltz

```
(
  pt_panel_id           INTEGER REFERENCES panel(p_panel_id) NOT FOR REPLICATION,  -- panel ID
  pt_tz_0               INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_1               INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_2               INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_3               INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_4               INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_5               INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_6               INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_7               INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_8               INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_9               INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_10              INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_11              INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_12              INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_13              INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_14              INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_15              INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_16              INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_17              INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
  pt_tz_18              INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION, -- timezone ID (NULL if empty)
```

```
pt_tz_19        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_20        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_21        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_22        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_23        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_24        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_25        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_26        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_27        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_28        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_29        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_30        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_31        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_32        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_33        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_34        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_35        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_36        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_37        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_38        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_39        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_40        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_41        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_42        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_43        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_44        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_45        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_46        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_47        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_48        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_49        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_50        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_51        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_52        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_53        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_54        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_55        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_56        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_57        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_58        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_59        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_60        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_61        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_62        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_tz_63        INTEGER NULL REFERENCES timezone(tz_id) NOT FOR REPLICATION,        -- timezone ID (NULL if empty)
pt_og_0         INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_1         INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_2         INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_3         INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_4         INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_5         INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_6         INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_7         INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_8         INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_9         INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_10        INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_11        INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_12        INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_13        INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_14        INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_15        INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_16        INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_17        INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_18        INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
pt_og_19        INTEGER NULL REFERENCES outputgrp(op_group_id)  NOT FOR REPLICATION, -- output group ID (NULL if empty)
```

```
  pt_og_20          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_21          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_22          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_23          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_24          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_25          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_26          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_27          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_28          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_29          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_30          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_31          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_32          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_33          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_34          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_35          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_36          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_37          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_38          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_39          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_40          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_41          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_42          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_43          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_44          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_45          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_46          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_47          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_48          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_49          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_50          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_51          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_52          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_53          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_54          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_55          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_56          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_57          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_58          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_59          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_60          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_61          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_62          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  pt_og_63          INTEGER NULL REFERENCES outputgrp(op_group_id) NOT FOR REPLICATION, -- output group ID (NULL if empty)
  site              CHAR(32) NOT NULL,                                                 -- Site Name
  pt_guid           uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                       -- Global unique record identifier
  pt_checksum       INTEGER NOT NULL,                                                  -- Record checksum
  PRIMARY KEY       (pt_panel_id)
);
```

CREATE TABLE paneltz_save

```
(
  pt_panel_id       INTEGER NOT NULL,  -- panel ID
  pt_tz_0           INTEGER NULL,      -- timezone ID (NULL if empty)
  pt_tz_1           INTEGER NULL,      -- timezone ID (NULL if empty)
  pt_tz_2           INTEGER NULL,      -- timezone ID (NULL if empty)
  pt_tz_3           INTEGER NULL,      -- timezone ID (NULL if empty)
  pt_tz_4           INTEGER NULL,      -- timezone ID (NULL if empty)
  pt_tz_5           INTEGER NULL,      -- timezone ID (NULL if empty)
  pt_tz_6           INTEGER NULL,      -- timezone ID (NULL if empty)
  pt_tz_7           INTEGER NULL,      -- timezone ID (NULL if empty)
  pt_tz_8           INTEGER NULL,      -- timezone ID (NULL if empty)
  pt_tz_9           INTEGER NULL,      -- timezone ID (NULL if empty)
```

```
pt_tz_10        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_11        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_12        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_13        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_14        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_15        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_16        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_17        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_18        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_19        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_20        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_21        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_22        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_23        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_24        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_25        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_26        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_27        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_28        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_29        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_30        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_31        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_32        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_33        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_34        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_35        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_36        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_37        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_38        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_39        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_40        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_41        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_42        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_43        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_44        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_45        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_46        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_47        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_48        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_49        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_50        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_51        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_52        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_53        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_54        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_55        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_56        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_57        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_58        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_59        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_60        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_61        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_62        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_tz_63        INTEGER NULL,    -- timezone ID (NULL if empty)
pt_og_0         INTEGER NULL,    -- output group ID (NULL if empty)
pt_og_1         INTEGER NULL,    -- output group ID (NULL if empty)
pt_og_2         INTEGER NULL,    -- output group ID (NULL if empty)
pt_og_3         INTEGER NULL,    -- output group ID (NULL if empty)
pt_og_4         INTEGER NULL,    -- output group ID (NULL if empty)
pt_og_5         INTEGER NULL,    -- output group ID (NULL if empty)
pt_og_6         INTEGER NULL,    -- output group ID (NULL if empty)
pt_og_7         INTEGER NULL,    -- output group ID (NULL if empty)
pt_og_8         INTEGER NULL,    -- output group ID (NULL if empty)
pt_og_9         INTEGER NULL,    -- output group ID (NULL if empty)
pt_og_10        INTEGER NULL,    -- output group ID (NULL if empty)
```

```
    pt_og_11            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_12            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_13            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_14            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_15            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_16            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_17            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_18            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_19            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_20            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_21            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_22            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_23            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_24            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_25            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_26            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_27            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_28            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_29            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_30            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_31            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_32            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_33            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_34            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_35            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_36            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_37            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_38            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_39            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_40            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_41            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_42            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_43            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_44            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_45            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_46            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_47            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_48            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_49            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_50            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_51            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_52            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_53            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_54            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_55            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_56            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_57            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_58            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_59            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_60            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_61            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_62            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    pt_og_63            INTEGER NULL,                                                   -- output group ID (NULL if empty)
    site                CHAR(32) NOT NULL,                                              -- Site Name
    pt_guid             uniqueidentifier NOT NULL,                                      -- Global unique record identifier
    pt_checksum         INTEGER NOT NULL,                                               -- Record checksum
    upd_guid            uniqueidentifier NOT NULL,                                      -- Guid for update operation
    upd_mode            SMALLINT NOT NULL,                                              -- Update mode (0 - delete, 1 - edit, 2 - new)
    upd_timestamp       DATETIME NOT NULL,                                              -- Date / Time of the update
    upd_checksum        INTEGER NOT NULL,                                               -- Checksum
    upd_rowguid         uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,    -- Global unique record identifier
    PRIMARY KEY         (pt_panel_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE panel_assaabloy

```
(
  paa_dsr_id            INTEGER REFERENCES IntegrationComponent(igc_id) NOT NULL,
  paa_panel_id          INTEGER REFERENCES Panel(p_panel_id) NOT NULL,
  paa_serial_number     char(32) NOT NULL,
  paa_tz_id_0           INTEGER NULL,
  paa_tz_id_1           INTEGER NULL,
  paa_tz_id_2           INTEGER NULL,
  paa_tz_id_3           INTEGER NULL,
  paa_tz_id_4           INTEGER NULL,
  paa_tz_id_5           INTEGER NULL,
  paa_tz_id_6           INTEGER NULL,
  paa_tz_id_7           INTEGER NULL,
  paa_tz_id_8           INTEGER NULL,
  paa_tz_id_9           INTEGER NULL,
  paa_tz_id_10          INTEGER NULL,
  paa_tz_id_11          INTEGER NULL,
  paa_tz_id_12          INTEGER NULL,
  paa_tz_id_13          INTEGER NULL,
  paa_tz_id_14          INTEGER NULL,
  paa_tz_id_15          INTEGER NULL,
  paa_tz_id_16          INTEGER NULL,
  paa_tz_id_17          INTEGER NULL,
  paa_tz_id_18          INTEGER NULL,
  paa_tz_id_19          INTEGER NULL,
  paa_tz_id_20          INTEGER NULL,
  paa_tz_id_21          INTEGER NULL,
  paa_tz_id_22          INTEGER NULL,
  paa_tz_id_23          INTEGER NULL,
  paa_tz_id_24          INTEGER NULL,
  paa_tz_id_25          INTEGER NULL,
  paa_tz_id_26          INTEGER NULL,
  paa_tz_id_27          INTEGER NULL,
  paa_tz_id_28          INTEGER NULL,
  paa_tz_id_29          INTEGER NULL,
  paa_tz_id_30          INTEGER NULL,
  paa_tz_id_31          INTEGER NULL,
  paa_schedule_0        UNIQUEIDENTIFIER NULL,
  paa_schedule_1        UNIQUEIDENTIFIER NULL,
  paa_schedule_2        UNIQUEIDENTIFIER NULL,
  paa_schedule_3        UNIQUEIDENTIFIER NULL,
  paa_schedule_4        UNIQUEIDENTIFIER NULL,
  paa_schedule_5        UNIQUEIDENTIFIER NULL,
  paa_schedule_6        UNIQUEIDENTIFIER NULL,
  paa_schedule_7        UNIQUEIDENTIFIER NULL,
  paa_schedule_8        UNIQUEIDENTIFIER NULL,
  paa_schedule_9        UNIQUEIDENTIFIER NULL,
  paa_schedule_10       UNIQUEIDENTIFIER NULL,
  paa_schedule_11       UNIQUEIDENTIFIER NULL,
  paa_schedule_12       UNIQUEIDENTIFIER NULL,
  paa_schedule_13       UNIQUEIDENTIFIER NULL,
  paa_schedule_14       UNIQUEIDENTIFIER NULL,
  paa_schedule_15       UNIQUEIDENTIFIER NULL,
  paa_schedule_16       UNIQUEIDENTIFIER NULL,
  paa_schedule_17       UNIQUEIDENTIFIER NULL,
  paa_schedule_18       UNIQUEIDENTIFIER NULL,
  paa_schedule_19       UNIQUEIDENTIFIER NULL,
  paa_schedule_20       UNIQUEIDENTIFIER NULL,
  paa_schedule_21       UNIQUEIDENTIFIER NULL,
  paa_schedule_22       UNIQUEIDENTIFIER NULL,
  paa_schedule_23       UNIQUEIDENTIFIER NULL,
  paa_schedule_24       UNIQUEIDENTIFIER NULL,
  paa_schedule_25       UNIQUEIDENTIFIER NULL,
  paa_schedule_26       UNIQUEIDENTIFIER NULL,
```

```
    paa_schedule_27         UNIQUEIDENTIFIER NULL,
    paa_schedule_28         UNIQUEIDENTIFIER NULL,
    paa_schedule_29         UNIQUEIDENTIFIER NULL,
    paa_schedule_30         UNIQUEIDENTIFIER NULL,
    paa_schedule_31         UNIQUEIDENTIFIER NULL,
    site                    [char](32) NOT NULL,
    paa_guid                [uniqueidentifier] ROWGUIDCOL NOT NULL,
    paa_checksum            INTEGER NOT NULL,                                   -- Record checksum
    PRIMARY KEY             (paa_panel_id)
)
```

CREATE TABLE panel_assaabloy_save

```
(
    paa_dsr_id              INTEGER NOT NULL,
    paa_panel_id            INTEGER NOT NULL,
    paa_serial_number       char(32) NOT NULL,
    paa_tz_id_0             INTEGER NULL,
    paa_tz_id_1             INTEGER NULL,
    paa_tz_id_2             INTEGER NULL,
    paa_tz_id_3             INTEGER NULL,
    paa_tz_id_4             INTEGER NULL,
    paa_tz_id_5             INTEGER NULL,
    paa_tz_id_6             INTEGER NULL,
    paa_tz_id_7             INTEGER NULL,
    paa_tz_id_8             INTEGER NULL,
    paa_tz_id_9             INTEGER NULL,
    paa_tz_id_10            INTEGER NULL,
    paa_tz_id_11            INTEGER NULL,
    paa_tz_id_12            INTEGER NULL,
    paa_tz_id_13            INTEGER NULL,
    paa_tz_id_14            INTEGER NULL,
    paa_tz_id_15            INTEGER NULL,
    paa_tz_id_16            INTEGER NULL,
    paa_tz_id_17            INTEGER NULL,
    paa_tz_id_18            INTEGER NULL,
    paa_tz_id_19            INTEGER NULL,
    paa_tz_id_20            INTEGER NULL,
    paa_tz_id_21            INTEGER NULL,
    paa_tz_id_22            INTEGER NULL,
    paa_tz_id_23            INTEGER NULL,
    paa_tz_id_24            INTEGER NULL,
    paa_tz_id_25            INTEGER NULL,
    paa_tz_id_26            INTEGER NULL,
    paa_tz_id_27            INTEGER NULL,
    paa_tz_id_28            INTEGER NULL,
    paa_tz_id_29            INTEGER NULL,
    paa_tz_id_30            INTEGER NULL,
    paa_tz_id_31            INTEGER NULL,
    paa_schedule_0          UNIQUEIDENTIFIER NULL,
    paa_schedule_1          UNIQUEIDENTIFIER NULL,
    paa_schedule_2          UNIQUEIDENTIFIER NULL,
    paa_schedule_3          UNIQUEIDENTIFIER NULL,
    paa_schedule_4          UNIQUEIDENTIFIER NULL,
    paa_schedule_5          UNIQUEIDENTIFIER NULL,
    paa_schedule_6          UNIQUEIDENTIFIER NULL,
    paa_schedule_7          UNIQUEIDENTIFIER NULL,
    paa_schedule_8          UNIQUEIDENTIFIER NULL,
    paa_schedule_9          UNIQUEIDENTIFIER NULL,
    paa_schedule_10         UNIQUEIDENTIFIER NULL,
    paa_schedule_11         UNIQUEIDENTIFIER NULL,
    paa_schedule_12         UNIQUEIDENTIFIER NULL,
    paa_schedule_13         UNIQUEIDENTIFIER NULL,
    paa_schedule_14         UNIQUEIDENTIFIER NULL,
```

```
    paa_schedule_15         UNIQUEIDENTIFIER NULL,
    paa_schedule_16         UNIQUEIDENTIFIER NULL,
    paa_schedule_17         UNIQUEIDENTIFIER NULL,
    paa_schedule_18         UNIQUEIDENTIFIER NULL,
    paa_schedule_19         UNIQUEIDENTIFIER NULL,
    paa_schedule_20         UNIQUEIDENTIFIER NULL,
    paa_schedule_21         UNIQUEIDENTIFIER NULL,
    paa_schedule_22         UNIQUEIDENTIFIER NULL,
    paa_schedule_23         UNIQUEIDENTIFIER NULL,
    paa_schedule_24         UNIQUEIDENTIFIER NULL,
    paa_schedule_25         UNIQUEIDENTIFIER NULL,
    paa_schedule_26         UNIQUEIDENTIFIER NULL,
    paa_schedule_27         UNIQUEIDENTIFIER NULL,
    paa_schedule_28         UNIQUEIDENTIFIER NULL,
    paa_schedule_29         UNIQUEIDENTIFIER NULL,
    paa_schedule_30         UNIQUEIDENTIFIER NULL,
    paa_schedule_31         UNIQUEIDENTIFIER NULL,
    site                    [char](32) NOT NULL,
    paa_guid                uniqueidentifier NOT NULL,
    paa_checksum            INTEGER NOT NULL,
    upd_guid                uniqueidentifier NOT NULL,          -- Guid for update operation
    upd_mode                SMALLINT NOT NULL,                  -- Update mode (0 - delete, 1 - edit, 2 - new)
    upd_timestamp           DATETIME NOT NULL,                  -- Date / Time of the update
    upd_checksum            INTEGER NOT NULL,                   -- Checksum
    upd_rowguid             uniqueidentifier ROWGUIDCOL DEFAULT
                                newid() UNIQUE NOT NULL,        -- Global unique record identifier
    PRIMARY KEY             ( paa_panel_id, upd_mode desc, upd_guid )
)
```

CREATE TABLE panel_ck

```
(
    p_id                                INTEGER REFERENCES panel(p_panel_id) NOT FOR REPLICATION,--
    p_trans_type                        SMALLINT,               -- reserved for legacy panels
    p_scram_code                        SMALLINT,               -- scramble mode 0-7
    p_system_override                   SMALLINT,               -- 1 = system override
    p_latch_alm_output                  SMALLINT,               -- 1 = latch alarm output
    p_group_outputs                     SMALLINT,               -- 1 = group outputs
    p_rest_hist                         SMALLINT,               -- 1 = restrict history storage
    p_rest_hist_days                    SMALLINT,               -- number of days after which history is deleted
    p_rest_hist_time                    DATETIME,               -- time at which to delete history
    p_wiegand_fac                       INTEGER,                -- facility code for wiegand cards (legacy panels only)
    p_ncrypt_fac                        INTEGER,                -- facility code for ncrypt cards (legacy panels only)
    p_mag_fac                           INTEGER,                -- facility code for mag stripe cards (legacy panels only)
    p_ovr_tailgate                      SMALLINT,               -- 1 = timed override/anti tailgate
    p_eci_unit_num                      INTEGER,                -- D600E eci unit with which this panel communicates
    p_high_speed_485                    SMALLINT NOT NULL,      -- Use high speed serial in CK720 Ver. 2 and higher
    p_hist_upload_threshold_flag        SMALLINT NOT NULL,      -- 1 = Turn on upload threshold, 0 = Turn off
    p_hist_upload_threshold             SMALLINT NOT NULL,      -- Threshold percent
    p_hist_always_upload_threshold_flag SMALLINT NOT NULL,      -- 1 = Turn on always upload threshold, 0 = Turn off
    p_hist_always_upload_threshold      SMALLINT NOT NULL,      -- Threshold percent
    p_hist_delete_flag                  SMALLINT NOT NULL,      -- 1 = Turn on history retention limit, 0 = Turn off
    p_hist_delete_days                  SMALLINT NOT NULL,      -- History retention limit days
    p_hist_delete_time                  INTEGER NOT NULL,       -- History retention limit time of day
    p_hist_upload_timezone              INTEGER NULL REFERENCES timezone(tz_id)
                                            NOT FOR REPLICATION,-- Timezone for history uploads
    p_facility_code4                    INTEGER,                -- facility codeD600-AP
    p_security_level                    SMALLINT,               -- (0-99) representing threat level
    p_mag_stripe_1                      SMALLINT,               --mag stripe for the readers
    p_mag_stripe_2                      SMALLINT,               --mag stripe for the readers
    p_mag_stripe_3                      SMALLINT,               --mag stripe for the readers
    p_mag_stripe_4                      SMALLINT,               --mag stripe for the readers
    p_mag_stripe_5                      SMALLINT,               --mag stripe for the readers
    p_mag_stripe_6                      SMALLINT,               --mag stripe for the readers
```

```
  p_mag_stripe_7                    SMALLINT,                          --mag stripe for the readers
  p_mag_stripe_8                    SMALLINT,                          --mag stripe for the readers
  p_mag_stripe_9                    SMALLINT,                          --mag stripe for the readers
  p_mag_stripe_10                   SMALLINT,                          --mag stripe for the readers
  p_duress_plus1                    SMALLINT,                          --
  p_log_rd_strike_msg               SMALLINT,                          --
  p_log_op_status                   SMALLINT,                          --
  p_enable_pin_duress               SMALLINT,                          --
  p_to_entry_require                SMALLINT,                          --
  p_num_pin_code_retry              SMALLINT,                          --
  p_elev_type                       INT NOT NULL,                      -- elevator type
  p_elev_param_1                    INT NOT NULL,                      -- High Level Elevator Parmeter, for KONE, Baudrate
  p_elev_param_2                    INT NOT NULL,                      -- High Level Elevator Parmeter, KONE Group
                                                                          Controller Address (1 to 8)
  p_elev_param_3                    INT NOT NULL,                      -- High Level Elevator Parmeter, for KONE, lowest
                                                                          level for group (1 to 128)
  site                              CHAR(32) NOT NULL,                 -- Site Name
  p_guid                            uniqueidentifier ROWGUIDCOL
                                       UNIQUE NOT NULL,                -- Global unique record identifier
  p_checksum                        INTEGER NOT NULL,                  -- Record checksum
  p_no_badge_archive                SMALLINT NOT NULL,
  p_no_accgrp_archive               SMALLINT NOT NULL,
  p_no_config_archive               SMALLINT NOT NULL,
  p_enable_suppress_msg             SMALLINT NOT NULL,
  p_custom_configuration            INTEGER NOT NULL,
  p_panel_initiated_sync_broadcast  SMALLINT NOT NULL,
  p_sync_broadcast_port             INTEGER NOT NULL,
  p_db_backup_period                INTEGER NOT NULL,
  PRIMARY KEY                       (p_id)
);
```

CREATE TABLE panel_ck_save

```
(
  p_id                              INTEGER NOT NULL,                  --
  p_trans_type                      SMALLINT,                          -- reserved for legacy panels
  p_scram_code                      SMALLINT,                          -- scramble mode 0-7
  p_system_override                 SMALLINT,                          -- 1 = system override
  p_latch_alm_output                SMALLINT,                          -- 1 = latch alarm output
  p_group_outputs                   SMALLINT,                          -- 1 = group outputs
  p_rest_hist                       SMALLINT,                          -- 1 = restrict history storage
  p_rest_hist_days                  SMALLINT,                          -- number of days after which history is deleted
  p_rest_hist_time                  DATETIME,                          -- time at which to delete history
  p_wiegand_fac                     INTEGER,                           -- facility code for wiegand cards (legacy panels only)
  p_ncrypt_fac                      INTEGER,                           -- facility code for ncrypt cards (legacy panels only)
  p_mag_fac                         INTEGER,                           -- facility code for mag stripe cards (legacy panels only)
  p_ovr_tailgate                    SMALLINT,                          -- 1 = timed override/anti tailgate
  p_eci_unit_num                    INTEGER,                           -- D600E eci unit with which this panel communicates
  p_high_speed_485                  SMALLINT NOT NULL,                 -- Use high speed serial in CK720 Ver. 2 and higher
  p_hist_upload_threshold_flag      SMALLINT NOT NULL,                 -- 1 = Turn on upload threshold, 0 = Turn off
  p_hist_upload_threshold           SMALLINT NOT NULL,                 -- Threshold percent
  p_hist_always_upload_threshold_flag SMALLINT NOT NULL,               -- 1 = Turn on always upload threshold, 0 = Turn off
  p_hist_always_upload_threshold    SMALLINT NOT NULL,                 -- Threshold percent
  p_hist_delete_flag                SMALLINT NOT NULL,                 -- 1 = Turn on history retention limit, 0 = Turn off
  p_hist_delete_days                SMALLINT NOT NULL,                 -- History retention limit days
  p_hist_delete_time                INTEGER NOT NULL,                  -- History retention limit time of day
  p_hist_upload_timezone            INTEGER NULL,                      -- Timezone for history uploads
  p_facility_code4                  INTEGER,                           -- facility codeD600-AP
  p_security_level                  SMALLINT,                          -- (0-99) representing threat level
  p_mag_stripe_1                    SMALLINT,                          --mag stripe for the readers
  p_mag_stripe_2                    SMALLINT,                          --mag stripe for the readers
  p_mag_stripe_3                    SMALLINT,                          --mag stripe for the readers
  p_mag_stripe_4                    SMALLINT,                          --mag stripe for the readers
  p_mag_stripe_5                    SMALLINT,                          --mag stripe for the readers
```

```
  p_mag_stripe_6                SMALLINT,                       --mag stripe for the readers
  p_mag_stripe_7                SMALLINT,                       --mag stripe for the readers
  p_mag_stripe_8                SMALLINT,                       --mag stripe for the readers
  p_mag_stripe_9                SMALLINT,                       --mag stripe for the readers
  p_mag_stripe_10               SMALLINT,                       --mag stripe for the readers
  p_duress_plus1                SMALLINT,                       --
  p_log_rd_strike_msg           SMALLINT,                       --
  p_log_op_status               SMALLINT,                       --
  p_enable_pin_duress           SMALLINT,                       --
  p_to_entry_require            SMALLINT,                       --
  p_num_pin_code_retry          SMALLINT,                       --
  p_elev_type                   INT NOT NULL,                   -- elevator type
  p_elev_param_1                INT NOT NULL,                   -- High Level Elevator Parmeter, for KONE, Baudrate
  p_elev_param_2                INT NOT NULL,                   -- High Level Elevator Parmeter, KONE Group
                                                                   Controller Address (1 to 8)
  p_elev_param_3                INT NOT NULL,                   -- High Level Elevator Parmeter, for KONE, lowest
                                                                   level for group (1 to 128)
  site                          CHAR(32) NOT NULL,              -- Site Name
  p_guid                        uniqueidentifier NOT NULL,      -- Global unique record identifier
  p_checksum                    INTEGER NOT NULL,               -- Record checksum
  upd_guid                      uniqueidentifier NOT NULL,      -- Guid for update operation
  upd_mode                      SMALLINT NOT NULL,              -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                 DATETIME NOT NULL,              -- Date / Time of the update
  upd_checksum                  INTEGER NOT NULL,               -- Checksum
  upd_rowguid                   uniqueidentifier ROWGUIDCOL DEFAULT
                                   newid() UNIQUE NOT NULL,     -- Global unique record identifier
  p_no_badge_archive            SMALLINT,
  p_no_accgrp_archive           SMALLINT,
  p_no_config_archive           SMALLINT,
  p_enable_suppress_msg         SMALLINT,
  p_custom_configuration        INTEGER,
  p_panel_initiated_sync_broadcast SMALLINT,
  p_sync_broadcast_port         INTEGER,
  p_db_backup_period            INTEGER,
  PRIMARY KEY                   (p_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE panel_hid

```
(
  phid_id                       INTEGER REFERENCES panel(p_panel_id)
                                   NOT FOR REPLICATION,         -- Unique database ID of panel
  phid_mac_address              CHAR(16) UNIQUE NOT NULL,       -- mac address of panel
  phid_heartbeat_xmit_interval  INTEGER NOT NULL,               -- Downloaded to panel for heartbeat timer
  phid_no_reception_timeout     INTEGER NOT NULL,               -- Time in seconds to wait for a msg from panel
  phid_resend_attempt_interval  INTEGER NOT NULL,               -- Downloaded to panel for its resend interval
  phid_utc_offset               INTEGER NOT NULL,               -- UTC offset
  phid_dst_used_flag            SMALLINT NOT NULL,              -- 1 = Daylight savings used by panel, 0 = not used
  phid_dst_adjustment           INTEGER NOT NULL,               -- Minutes, daylight savings adjustment, this is always an hour
  phid_dst_begin_month          SMALLINT NOT NULL,              -- Month daylight savings begins (1 - 12)
  phid_dst_begin_week           SMALLINT NOT NULL,              -- Week of month daylight savings begins (1 - 5)
  phid_dst_begin_day            SMALLINT NOT NULL,              -- Day of week daylight savings begins (1 - 7)
  phid_dst_begin_time           INT NOT NULL,                   -- Time of day daylight savings begins (seconds)
  phid_dst_end_month            SMALLINT NOT NULL,              -- Month daylight savings end (1 - 12)
  phid_dst_end_week             SMALLINT NOT NULL,              -- Week of month daylight savings end (1 - 5)
  phid_dst_end_day              SMALLINT NOT NULL,              -- Day of week daylight savings end (1 - 7)
  phid_dst_end_time             INT NOT NULL,                   -- Time of day daylight savings end (seconds)
  phid_encryption_key_changed   SMALLINT NOT NULL,              -- 1 = encryption key changed
  site                          CHAR(32) NOT NULL,              -- Site Name
  phid_guid                     uniqueidentifier ROWGUIDCOL
                                   UNIQUE NOT NULL,             -- Global unique record identifier
  phid_checksum                 INTEGER NOT NULL,               -- Record checksum
  PRIMARY KEY                   (phid_id)
);
```

CREATE TABLE panel_hid_save

```
(
  phid_id                       INTEGER NOT NULL,                    -- Unique database ID of panel
  phid_mac_address              CHAR(16) NOT NULL,                   -- optional, mac address of panel
  phid_heartbeat_xmit_interval  INTEGER NOT NULL,                    -- Downloaded to panel for heartbeat timer
  phid_no_reception_timeout     INTEGER NOT NULL,                    -- Time in seconds to wait for a msg from panel
  phid_resend_attempt_interval  INTEGER NOT NULL,                    -- Downloaded to panel for its resend interval
  phid_utc_offset               INTEGER NOT NULL,                    -- UTC offset
  phid_dst_used_flag            SMALLINT NOT NULL,                   -- 1 = Daylight savings used by panel, 0 = not used
  phid_dst_adjustment           INTEGER NOT NULL,                    -- Minutes, daylight savings adjustment, this is always an hour
  phid_dst_begin_month          SMALLINT NOT NULL,                   -- Month daylight savings begins
  phid_dst_begin_week           SMALLINT NOT NULL,                   -- Week of month daylight savings begins
  phid_dst_begin_day            SMALLINT NOT NULL,                   -- Day of week daylight savings begins
  phid_dst_begin_time           SMALLINT NOT NULL,                   -- Time of day daylight savings begins
  phid_dst_end_month            SMALLINT NOT NULL,                   -- Month daylight savings end
  phid_dst_end_week             SMALLINT NOT NULL,                   -- Week of month daylight savings end
  phid_dst_end_day              SMALLINT NOT NULL,                   -- Day of week daylight savings end
  phid_dst_end_time             SMALLINT NOT NULL,                   -- Time of day daylight savings end
  phid_encryption_key_changed   SMALLINT NOT NULL,                   -- 1 = encrytion key changed
  site                          CHAR(32) NOT NULL,                   -- Site Name
  phid_guid                     uniqueidentifier NOT NULL,           -- Global unique record identifier
  phid_checksum                 INTEGER NOT NULL,                    -- Record checksum
  upd_guid                      uniqueidentifier NOT NULL,           -- Guid for update operation
  upd_mode                      SMALLINT NOT NULL,                   -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                 DATETIME NOT NULL,                   -- Date / Time of the update
  upd_checksum                  INTEGER NOT NULL,                    -- Checksum
  upd_rowguid                   uniqueidentifier ROWGUIDCOL DEFAULT newid()
                                    UNIQUE NOT NULL,                 -- Global unique record identifier
  PRIMARY KEY                   (phid_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE panel_osi

```
(
  posi_id          INTEGER REFERENCES panel(p_panel_id) NOT FOR REPLICATION,  -- Unique database ID of panel
  posi_mac         CHAR(16) UNIQUE NOT NULL,                                 -- mac address of panel
  site             CHAR(32) NOT NULL,                                        -- Site Name
  posi_guid        uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,              -- Global unique record identifier
  posi_checksum    INTEGER NOT NULL,                                         -- Record checksum
  posi_channels    INTEGER NOT NULL,                                         -- Record channels
  PRIMARY KEY      (posi_id)
);
```

CREATE TABLE panel_osi_save

```
(
  posi_id          INTEGER NOT NULL,                                         -- Unique database ID of panel
  posi_mac         CHAR(16) NOT NULL,                                        -- mac address of panel
  site             CHAR(32) NOT NULL,                                        -- Site Name
  posi_guid        uniqueidentifier NOT NULL,                                -- Global unique record identifier
  posi_checksum    INTEGER NOT NULL,                                         -- Record checksum
  posi_channels    INTEGER ,                                                 -- Record channels
  upd_guid         uniqueidentifier NOT NULL,                                -- Guid for update operation
  upd_mode         SMALLINT NOT NULL,                                        -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp    DATETIME NOT NULL,                                        -- Date / Time of the update
  upd_checksum     INTEGER NOT NULL,                                         -- Checksum
  upd_rowguid      uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,  -- Global unique record identifier
  PRIMARY KEY      (posi_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE panel_p900

```
(
  pcp_id              INTEGER REFERENCES panel(p_panel_id) NOT FOR REPLICATION,   -- Unique database ID of panel
  site                CHAR(32) NOT NULL,                                         -- Site Name
  pcp_guid            uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,               -- Global unique record identifier
  pcp_checksum        INTEGER NOT NULL,                                          -- Record checksum
  PRIMARY KEY         (pcp_id)
);
```

CREATE TABLE panel_p900_save

```
(
  pcp_id              INTEGER  NOT NULL,                                         -- Unique database ID of panel
  site                CHAR(32) NOT NULL,                                         -- Site Name
  pcp_guid            uniqueidentifier NOT NULL,                                 -- Global unique record identifier
  pcp_checksum        INTEGER NOT NULL,                                          -- Record checksum
  upd_guid            uniqueidentifier NOT NULL,                                 -- Guid for update operation
  upd_mode            SMALLINT NOT NULL,                                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp       DATETIME NOT NULL,                                         -- Date / Time of the update
  upd_checksum        INTEGER NOT NULL,                                          -- Checksum
  upd_rowguid         uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY         (pcp_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE panel_s321ip

```
(
  ps321ip_id                  INTEGER REFERENCES panel(p_panel_id)
                                NOT FOR REPLICATION,                  -- Unique database ID of panel
  ps321ip_enable_panel_inputs SMALLINT NOT NULL,                      -- enable the 2 panel input points
  ps321ip_mac_address         CHAR(16) UNIQUE NOT NULL,               -- mac address of panel
  ps321ip_dns_lookup_name     CHAR(255),                              -- DNS name used to resolve panel ip address
  ps321ip_use_dns_not_ip_flag SMALLINT NOT NULL,                      -- 1 = use DNS lookup, 0 = use ip address
  ps321ip_snmp_enable_flag    SMALLINT NOT NULL,                      -- 1 = enable SNMP interface on panel, 0 = disable
  ps321ip_web_ui_enable_flag  SMALLINT NOT NULL,                      -- 1 = enable Web UI interface on panel, 0 = disable
  ps321ip_heartbeat_xmit_interval INTEGER NOT NULL,                   -- Downloaded to panel for heartbeat timer
  ps321ip_no_reception_timeout INTEGER NOT NULL,                      -- Time in seconds to wait for a msg from panel
  ps321ip_resend_attempt_interval INTEGER NOT NULL,                   -- Downloaded to panel for its resend interval
  ps321ip_disconnect_delay    INTEGER NOT NULL,                       -- Downloaded to panel for its disconnect delay
  ps321ip_disconnect_delay_flag SMALLINT NOT NULL,                    -- 0 = Time Delay, 1 = Never, 2 = Immediate (only used by UI)
  ps321ip_hist_delete_flag    SMALLINT NOT NULL,                      -- 1 = Turn on history retention limit, 0 = Turn off
  ps321ip_hist_delete_days    SMALLINT NOT NULL,                      -- History retention limit days
  ps321ip_utc_offset          INTEGER NOT NULL,                       -- UTC offset
  ps321ip_dst_used_flag       SMALLINT NOT NULL,                      -- 1 = Daylight savings used by panel, 0 = not used
  ps321ip_dst_adjustment      INTEGER NOT NULL,                       -- Minutes, daylight savings adjustment, this is always an hour
  ps321ip_dst_begin_month     SMALLINT NOT NULL,                      -- Month daylight savings begins (1 - 12)
  ps321ip_dst_begin_week      SMALLINT NOT NULL,                      -- Week of month daylight savings begins (1 - 5)
  ps321ip_dst_begin_day       SMALLINT NOT NULL,                      -- Day of week daylight savings begins (1 - 7)
  ps321ip_dst_begin_time      INT NOT NULL,                           -- Time of day daylight savings begins (seconds)
  ps321ip_dst_end_month       SMALLINT NOT NULL,                      -- Month daylight savings end (1 - 12)
  ps321ip_dst_end_week        SMALLINT NOT NULL,                      -- Week of month daylight savings end (1 - 5)
  ps321ip_dst_end_day         SMALLINT NOT NULL,                      -- Day of week daylight savings end (1 - 7)
  ps321ip_dst_end_time        INT NOT NULL,                           -- Time of day daylight savings end (seconds)
  site                        CHAR(32) NOT NULL,                      -- Site Name
  ps321ip_guid                uniqueidentifier ROWGUIDCOL
                                UNIQUE NOT NULL,                      -- Global unique record identifier
  ps321ip_checksum            INTEGER NOT NULL,                       -- Record checksum
  PRIMARY KEY                 (ps321ip_id)
);
```

CREATE TABLE panel_s321ip_save

```
(
  ps321ip_id                    INTEGER NOT NULL,                  -- Unique database ID of panel
  ps321ip_enable_panel_inputs   SMALLINT NOT NULL,                 -- enable the 2 panel input points
  ps321ip_mac_address           CHAR(16) NOT NULL,                 -- optional, mac address of panel
  ps321ip_dns_lookup_name       CHAR(255),                         -- optional, DNS name used to resolve panel ip address
  ps321ip_use_dns_not_ip_flag   SMALLINT NOT NULL,                 -- 1 = use DNS lookup, 0 = use ip address
  ps321ip_snmp_enable_flag      SMALLINT NOT NULL,                 -- 1 = enable SNMP interface on panel, 0 = disable
  ps321ip_web_ui_enable_flag    SMALLINT NOT NULL,                 -- 1 = enable Web UI interface on panel, 0 = disable
  ps321ip_heartbeat_xmit_interval INTEGER NOT NULL,                -- Downloaded to panel for heartbeat timer
  ps321ip_no_reception_timeout  INTEGER NOT NULL,                  -- Time in seconds to wait for a msg from panel
  ps321ip_resend_attempt_interval INTEGER NOT NULL,                -- Downloaded to panel for its resend interval
  ps321ip_disconnect_delay      INTEGER NOT NULL,                  -- Downloaded to panel for its disconnect delay
  ps321ip_disconnect_delay_flag SMALLINT NOT NULL,                 -- 0 = Time Delay, 1 = Never, 2 = Immediate (only used by UI)
  ps321ip_hist_delete_flag      SMALLINT NOT NULL,                 -- 1 = Turn on history retention limit, 0 = Turn off
  ps321ip_hist_delete_days      SMALLINT NOT NULL,                 -- History retention limit days
  ps321ip_utc_offset            INTEGER NOT NULL,                  -- UTC offset
  ps321ip_dst_used_flag         SMALLINT NOT NULL,                 -- 1 = Daylight savings used by panel, 0 = not used
  ps321ip_dst_adjustment        INTEGER NOT NULL,                  -- Minutes, daylight savings adjustment, this is always an hour
  ps321ip_dst_begin_month       SMALLINT NOT NULL,                 -- Month daylight savings begins
  ps321ip_dst_begin_week        SMALLINT NOT NULL,                 -- Week of month daylight savings begins
  ps321ip_dst_begin_day         SMALLINT NOT NULL,                 -- Day of week daylight savings begins
  ps321ip_dst_begin_time        SMALLINT NOT NULL,                 -- Time of day daylight savings begins
  ps321ip_dst_end_month         SMALLINT NOT NULL,                 -- Month daylight savings end
  ps321ip_dst_end_week          SMALLINT NOT NULL,                 -- Week of month daylight savings end
  ps321ip_dst_end_day           SMALLINT NOT NULL,                 -- Day of week daylight savings end
  ps321ip_dst_end_time          SMALLINT NOT NULL,                 -- Time of day daylight savings end
  site                          CHAR(32) NOT NULL,                 -- Site Name
  ps321ip_guid                  uniqueidentifier NOT NULL,         -- Global unique record identifier
  ps321ip_checksum              INTEGER NOT NULL,                  -- Record checksum
  upd_guid                      uniqueidentifier NOT NULL,         -- Guid for update operation
  upd_mode                      SMALLINT NOT NULL,                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                 DATETIME NOT NULL,                 -- Date / Time of the update
  upd_checksum                  INTEGER NOT NULL,                  -- Checksum
  upd_rowguid                   uniqueidentifier ROWGUIDCOL DEFAULT
                                  newid() UNIQUE NOT NULL,         -- Global unique record identifier
  PRIMARY KEY                   (ps321ip_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE panel_save

```
(
  p_partition                   INTEGER,                           -- partition ID to which this panel belongs
  p_public                      SMALLINT,                          -- 1 = panel is public, 0 = panel is private
  p_unit_name                   VARCHAR(32) NOT NULL,              -- user entered name
  p_panel_id                    INTEGER NOT NULL,                  -- unique database ID
  p_query_string                CHAR(64) NULL,                     -- query string entered by user
  p_enabled                     SMALLINT NOT NULL,                 -- 1 = enabled
  p_code_major_version          SMALLINT,                          -- Panel firmware revision major portion
  p_code_minor_version          SMALLINT,                          -- Panel firmware revision minor portion
  p_code_build_version          SMALLINT,                          -- Panel firmware revision build portion
  p_ipl_major_version           SMALLINT,                          -- Panel ipl firmware revision major portion
  p_ipl_minor_version           SMALLINT,                          -- Panel ipl firmware revision minor portion
  p_ipl_build_version           SMALLINT,                          -- Panel ipl firmware revision build portion
  p_min_major_version           SMALLINT NOT NULL,                 -- Minimum version of panel firmware that can use this record
                                                                      major portion
  p_min_minor_version           SMALLINT NOT NULL,                 -- Minimum version of panel firmware that can use this record
                                                                      minor portion
  p_min_build_version           SMALLINT NOT NULL,                 -- Minimum version of panel firmware that can use this record
                                                                      build portion
  p_max_major_version           SMALLINT NOT NULL,                 -- Maximum version of panel firmware that can use this record
                                                                      major portion
  p_max_minor_version           SMALLINT NOT NULL,                 -- Maximum version of panel firmware that can use this record
                                                                      minor portion
```

```
  p_max_build_version          SMALLINT NOT NULL,          -- Maximum version of panel firmware that can use this record
                                                              build portion
  p_configuration_model        SMALLINT NOT NULL,          -- Panel model identifier 0 = Unknown, 1 = CK720, 2 = CK705,
                                                              3 = CK710
  p_hardware_model             SMALLINT NOT NULL,          -- Panel model identifier 0 = Unknown, 1 = CK720, 2 = CK705,
                                                              3 = CK710
  p_version_text               VARCHAR(70) NULL,           -- panel version string (set by comms)
  p_time_date_offset           INTEGER NOT NULL,           -- time offset in mins
  p_offset_enable              SMALLINT NOT NULL,          -- 1 = enable time offset
  p_ip_pri                     CHAR(16) NULL,              -- primary IP address
  p_ip_alt                     CHAR(16) NULL,              -- alternate IP address
  p_pri_comm_pref              SMALLINT NOT NULL,          -- primary comm is preferred path
  p_net_timeout                SMALLINT NOT NULL,          -- network timeout in secs
  p_lan_net_polling_interval   INTEGER NULL,               -- Downloaded to panel for primary channel polling
  p_dialup_net_polling_interval INTEGER NULL,              -- Downloaded to panel for alternate channel polling
  p_lan_net_polling_timeout    INTEGER NULL,               -- Host uses for primary channel polling
  p_dialup_net_polling_timeout INTEGER NULL,               -- Host uses for alternate channel polling
  p_unit                       INTEGER NOT NULL,           -- panel address
  p_poll_timeout               SMALLINT NOT NULL,          -- Time in milliseconds to wait for a poll response
  p_poll_direction             SMALLINT NOT NULL,          -- Preferred poll direction, 0 = forward, 1 = reverse
  p_actual_poll_dir            SMALLINT NOT NULL,          -- Actual poll direction, 0 = forward, 1 = reverse, -1 = unknown
  p_retry_delay                SMALLINT NOT NULL,          -- re establish delay for serial panels in seconds
  p_inp_report_delay           SMALLINT NOT NULL,          -- input reporting delay
  p_out_report_delay           SMALLINT NOT NULL,          -- output reporting delay
  p_tz_checking                SMALLINT NOT NULL,          -- 1 = timezone checking
  p_pin_type                   SMALLINT NOT NULL,          -- 0 = algorithmic, 1 = custom
  p_4_or_5_digit               SMALLINT NOT NULL,          -- 0 = 4 digit pin, 1 = 5 digit pin
  p_inxit                      SMALLINT NOT NULL,          -- 1 = entry/exit
  p_upload                     SMALLINT NOT NULL,          -- 1 = upload history
  site                         CHAR(32) NOT NULL,          -- Site Name
  p_guid                       uniqueidentifier NOT NULL,  -- Global unique record identifier
  p_checksum                   INTEGER NOT NULL,           -- Record checksum
  upd_guid                     uniqueidentifier NOT NULL,  -- Guid for update operation
  upd_mode                     SMALLINT NOT NULL,          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                DATETIME NOT NULL,          -- Date / Time of the update
  upd_checksum                 INTEGER NOT NULL,           -- Checksum
  upd_rowguid                  uniqueidentifier ROWGUIDCOL DEFAULT
                                 newid() UNIQUE NOT NULL,  -- Global unique record identifier
  p_bacnet                     SMALLINT NOT NULL,          -- 1 = bacnet useses this panel.
  p_loop_id                    INTEGER NULL,               -- loop number
  p_encryption_enabled         SMALLINT NOT NULL,          -- 1 = encryption enabled
  p_encryption_key             VARCHAR(256) NULL,          -- encryption key up to 256 characters
  PRIMARY KEY                  ( p_partition, p_panel_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE partition

```
(
  part_number                  INTEGER PRIMARY KEY,                          -- unique partition ID (Super User partition = 1)
  part_name                    CHAR(32) NOT NULL,                            -- partition name
  site                         CHAR(32) NOT NULL,                            -- Site Name
  part_guid                    uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  part_checksum                INTEGER NOT NULL,                             -- Record checksum
);
   CREATE UNIQUE INDEX         part_name ON partition ( part_name );
```

CREATE TABLE partition_save

```
(
  part_number                  INTEGER NOT NULL,           -- unique partition ID (Super User partition = 1)
  part_name                    CHAR(32) NOT NULL,          -- partition name
  site                         CHAR(32) NOT NULL,          -- Site Name
  part_guid                    uniqueidentifier NOT NULL,  -- Global unique record identifier
  part_checksum                INTEGER NOT NULL,           -- Record checksum
```

```
  upd_guid              uniqueidentifier NOT NULL,                                  -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                          -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                           -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                                          -- Global unique record identifier
  PRIMARY KEY           (part_number, upd_mode desc, upd_guid)
);
```

CREATE TABLE pin_number

```
(
  pn_id                 INTEGER IDENTITY NOT FOR REPLICATION PRIMARY KEY,   -- unique ID for this pin
  pn_name               CHAR(64) NOT NULL,                                  -- pin name
  pn_pin                VARCHAR(16) NOT NULL,                               -- the pin
  site                  CHAR(32) NOT NULL,                                  -- Site Name
  pn_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  pn_checksum           INTEGER NOT NULL,                                   -- Record checksum
);
  CREATE UNIQUE INDEX   pn_site_name ON pin_number (site, pn_name);
```

CREATE TABLE pin_number_save

```
(
  pn_id                 INTEGER NOT NULL,                                   -- unique ID for this pin
  pn_name               CHAR(64) NOT NULL,                                  -- pin name
  pn_pin                VARCHAR(16) NOT NULL,                               -- the pin
  site                  CHAR(32) NOT NULL,                                  -- Site Name
  pn_guid               uniqueidentifier NOT NULL,                          -- Global unique record identifier
  pn_checksum           INTEGER NOT NULL,                                   -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                          -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                  -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                   -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,  -- Global unique record identifier
  PRIMARY KEY           (pn_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE portconfig

```
(
  port_enum             INTEGER NOT NULL,                                   -- port name enumartion
  port_value            INTEGER NOT NULL,                                   -- port value
  site                  CHAR(32) NOT NULL,                                  -- Site Name
  port_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  port_checksum         INTEGER NOT NULL,                                   -- Record checksum
  PRIMARY KEY           (site, port_enum)
);
```

CREATE TABLE portconfig_save

```
(
  port_enum             INTEGER NOT NULL,                                   -- port name enumartion
  port_value            INTEGER NOT NULL,                                   -- port value
  site                  CHAR(32) NOT NULL,                                  -- Site Name
  port_guid             uniqueidentifier NOT NULL,                          -- Global unique record identifier
  port_checksum         INTEGER NOT NULL,                                   -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                          -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                  -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                   -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,  -- Global unique record identifier
  PRIMARY KEY           ( port_enum, site, upd_mode desc, upd_guid )
);
```

CREATE TABLE portrait

```
(
  pt_cardholder_id      INTEGER,                                          -- cardholder ID
  pt_create_date        DATETIME,                                         -- date image saved
  pt_image              IMAGE,                                            -- image data
  pt_instance           INTEGER,                                          -- portrait instance
  site                  CHAR(32) NOT NULL,                                -- Site Name
  pt_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,      -- Global unique record identifier
  pt_checksum           INTEGER NOT NULL,                                 -- Record checksum
  PRIMARY KEY           ( pt_cardholder_id, pt_instance )
);
```

CREATE TABLE portrait_save

```
(
  pt_cardholder_id      INTEGER,                                                    -- cardholder ID
  pt_create_date        DATETIME,                                                   -- date image saved
  pt_image              IMAGE,                                                      -- image data
  pt_instance           INTEGER,                                                    -- portrait instance
  site                  CHAR(32) NOT NULL,                                          -- Site Name
  pt_guid               uniqueidentifier NOT NULL,                                  -- Global unique record identifier
  pt_checksum           INTEGER NOT NULL,                                           -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                                  -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                          -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                           -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY           ( pt_cardholder_id, pt_instance, upd_mode desc, upd_guid )
);
```

CREATE TABLE privgrp

```
(
  pg_id                 INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,      -- unique menu privilege group ID
  pg_group_name         CHAR(32) NOT NULL,                                -- menu privilege group name
  site                  CHAR(32) NOT NULL,                                -- Site Name
  pg_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,      -- Global unique record identifier
  pg_checksum           INTEGER NOT NULL,                                 -- Record checksum
  PRIMARY KEY           ( pg_id)
);
  CREATE UNIQUE INDEX   pg_site_name ON privgrp (site, pg_group_name);
```

CREATE TABLE privgrpdata

```
(
  pd_group_id           INTEGER REFERENCES privgrp(pg_id)
                          NOT FOR REPLICATION,                            -- priv grp ID
  pd_function           SMALLINT NOT NULL,                                -- function identifier
-- 1 = Cardholder, 2 = Badge resync, 3 = Image recall,
   4 = Image recall filters, 5 = Alarm monitor,
   6 = Alarm instruction text, 7 = Instruction text,
   8 = Alarms filter, 9 = Alarms forwarding,
   10 = Configure events, 11 = Event counters,
   12 = Trigger events manually, 13 = Output control,
   14 = Panel relay, 15 = Door control, 16 = Control all doors,
   17 = Run report, 18 = Report configuration,
   19 = Assign operator, 20 = Change password,
   21 = Menu permission groups, 22 = Real time list,
   23 = Real time map, 24 = Download, 25 = Download status,
   26 = Service control, 27 = Workstation status,
   28 = System status, 29 = Database maintenance,
   30 = Update CK720 panels, 31 = CK720 write DB to flash,
   32 = Local, 33 = System, 34 = Company, 35 = Department,
   36 = Access template, 37 = User defined fields, 38 = Setup,
```

```
                                                    39 = Badge layout, 40 = Map maker, 41 = Icon editor,
                                                    42 = Badges, 43 = Cardholder images
                                                    10001 = Registration parameters, 10002 = Site parameters,
                                                    10003 = Workstation, 10004 = Partitions, 10005 = Counters,
                                                    10006 = Time zones, 10007 = Holidays,
                                                    10008 = Access groups, 10009 = Terminal groups,
                                                    10010 = Panel, 10011 = Panel timezones,
                                                    10012 = Panel holidays, 10013 = Soft alarm,
                                                    10014 = Input groups, 10015 = Output groups,
                                                    10016 = Panel card event, 10017 = Terminals,
                                                    10018 = Input points, 10019 = Output points,
                                                    10020 = Action interlocks, 10021 = BACnet sources
  pd_privilege          SMALLINT NOT NULL,          -- application privilege
                                                       0 = none, 1 = read, 2 = edit, 4 = full
  site                  CHAR(32) NOT NULL,          -- Site Name
  pd_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  pd_checksum           INTEGER NOT NULL,           -- Record checksum
  PRIMARY KEY           ( pd_group_id, pd_function)
);
  CREATE INDEX          pd_site ON privgrpdata ( site );
```

CREATE TABLE privgrpdata_save

```
(
  pd_group_id           INTEGER,                    -- priv grp ID
  pd_function           SMALLINT NOT NULL,          -- function identifier
                                                    1 = Cardholder, 2 = Badge resync, 3 = Image recall,
                                                    4 = Image recall filters, 5 = Alarm monitor,
                                                    6 = Alarm instruction text, 7 = Instruction text,
                                                    8 = Alarms filter, 9 = Alarms forwarding,
                                                    10 = Configure events, 11 = Event counters,
                                                    12 = Trigger events manually, 13 = Output control,
                                                    14 = Panel relay, 15 = Door control, 16 = Control all doors,
                                                    17 = Run report, 18 = Report configuration,
                                                    19 = Assign operator, 20 = Change password,
                                                    21 = Menu permission groups, 22 = Real time list,
                                                    23 = Real time map, 24 = Download, 25 = Download status,
                                                    26 = Service control, 27 = Workstation status,
                                                    28 = System status, 29 = Database maintenance,
                                                    30 = Update CK720 panels, 31 = CK720 write DB to flash,
                                                    32 = Local, 33 = System, 34 = Company, 35 = Department,
                                                    36 = Access template, 37 = User defined fields, 38 = Setup,
                                                    39 = Badge layout, 40 = Map maker, 41 = Icon editor,
                                                    42 = Badges, 43 = Cardholder images,
                                                    10001 = Registration parameters, 10002 = Site parameters,
                                                    10003 = Workstation, 10004 = Partitions, 10005 = Counters,
                                                    10006 = Time zones, 10007 = Holidays,
                                                    10008 = Access groups, 10009 = Terminal groups,
                                                    10010 = Panel, 10011 = Panel timezones,
                                                    10012 = Panel holidays, 10013 = Soft alarm,
                                                    10014 = Input groups, 10015 = Output groups,
                                                    10016 = Panel card event, 10017 = Terminals,
                                                    10018 = Input points, 10019 = Output points,
                                                    10020 = Action interlocks, 10021 = BACnet sources,
  pd_privilege          SMALLINT NOT NULL,          -- application privilege
                                                       0 = none, 1 = read, 2 = edit, 4 = full
  site                  CHAR(32) NOT NULL,          -- Site Name
  pd_guid               uniqueidentifier NOT NULL,  -- Global unique record identifier
  pd_checksum           INTEGER NOT NULL,           -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,  -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,          -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,           -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
```

```
                    UNIQUE NOT NULL,                    -- Global unique record identifier
  PRIMARY KEY       ( pd_group_id, pd_function, upd_mode desc, upd_guid)
);
```

CREATE TABLE privgrp_save

```
(
  pg_id             INTEGER NOT NULL,                                                  -- unique menu privilege group ID
  pg_group_name     CHAR(32) NOT NULL,                                                 -- menu privilege group name
  site              CHAR(32) NOT NULL,                                                 -- Site Name
  pg_guid           uniqueidentifier NOT NULL,                                         -- Global unique record identifier
  pg_checksum       INTEGER NOT NULL,                                                  -- Record checksum
  upd_guid          uniqueidentifier NOT NULL,                                         -- Guid for update operation
  upd_mode          SMALLINT NOT NULL,                                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp     DATETIME NOT NULL,                                                 -- Date / Time of the update
  upd_checksum      INTEGER NOT NULL,                                                  -- Checksum
  upd_rowguid       uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,       -- Global unique record identifier
  PRIMARY KEY       ( pg_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE processinfo

```
(
  pi_id             INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,
  pi_timestamp      DATETIME NOT NULL,
  pi_name           CHAR(64) NOT NULL,
  pi_process_id     INTEGER NOT NULL,
  pi_working_size   INTEGER NOT NULL,
  pi_peak_working_size INTEGER NOT NULL,
  PRIMARY KEY       (pi_id)
);
  CREATE INDEX      pi_name ON processinfo ( pi_name );
  CREATE INDEX      pi_timestamp ON processinfo ( pi_timestamp );
```

CREATE TABLE reason

```
(
  r_name            CHAR(32) PRIMARY KEY,                       -- Unique reason choice that can be assigned to a new badge
  site              CHAR(32) NOT NULL,                          -- Site Name
  r_guid            uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  r_checksum        INTEGER NOT NULL,                           -- Record checksum
);
  CREATE UNIQUE INDEX  r_site_name ON reason (site, r_name);
```

CREATE TABLE reason_save

```
(
  r_name            CHAR(32),                                   -- Unique reason choice that can be assigned to a new badge
  site              CHAR(32) NOT NULL,                          -- Site Name
  r_guid            uniqueidentifier NOT NULL,                  -- Global unique record identifier
  r_checksum        INTEGER NOT NULL,                           -- Record checksum
  upd_guid          uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode          SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp     DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum      INTEGER NOT NULL,                           -- Checksum
  upd_rowguid       uniqueidentifier ROWGUIDCOL DEFAULT newid()
                      UNIQUE NOT NULL,                          -- Global unique record identifier
  PRIMARY KEY       (r_name, upd_mode desc, upd_guid)
);
```

CREATE TABLE recipient

```
(
  r_id                INTEGER IDENTITY NOT FOR REPLICATION
                        PRIMARY KEY,                    -- recipient ID
  r_panel_id          INTEGER REFERENCES panel(p_panel_id)
                        NOT FOR REPLICATION,            -- panel ID
  r_category          INTEGER,                          -- 0 = Host Event Category, 1 = Host Log Category,
                                                          2 = Host Logic Category, 3 = Audit Log Category,
                                                          4 = Panel Category, 5 = Panel Event Category,
                                                          6 = Hardware Category, 7 = Terminal Category,
                                                          8 = Reader Category, 9 = Input Category,
                                                          10 = Output Category, 11 = Grant Category,
                                                          12 = Deny Category, 13 = Trace Category ,
                                                          14 = Time and Attendance Category,
  r_valid_days        TINYINT,                          -- Field from BACnet Notification Class
  r_from_time         DATETIME,                         -- Field from BACnet Notification Class
  r_to_time           DATETIME,                         -- Field from BACnet Notification Class
  r_choice            TINYINT,                          -- Field from BACnet Notification Class
  r_instance          INTEGER,                          -- Field from BACnet Notification Class
  r_network           SMALLINT,                         -- Field from BACnet Notification Class
  r_ip_addr           CHAR(16),                         -- Field from BACnet Notification Class
  r_udp_port          INTEGER,                          -- Field from BACnet Notification Class
  r_process           INTEGER,                          -- Field from BACnet Notification Class
  r_confirmed         BIT,                              -- Field from BACnet Notification Class
  r_transitions       TINYINT,                          -- Field from BACnet Notification Class
);
  CREATE INDEX        r_panel_id ON recipient ( r_panel_id );
  CREATE INDEX        r_category ON recipient ( r_category );
```

CREATE TABLE redn_xaction

```
(
  site                CHAR(32) NOT NULL,                -- Site Name
  redn_id             INTEGER IDENTITY NOT FOR REPLICATION
                        PRIMARY KEY,                    -- input ID to which this status belongs
  redn_guid           uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  redn_checksum       INTEGER NOT NULL,                 -- Record checksum
  redn_enable         SMALLINT NOT NULL,                -- 0: disable, 1: enable
  redn_action         SMALLINT NOT NULL,                -- failover action 0: no action, 1: system failover, 2: manual
                                                          failover, 3. failback, 4: config, 5: restarted, 6:status check
  redn_current_server SMALLINT NOT NULL,                -- working server 1: primary, 2 standby
  redn_pri_status     SMALLINT NOT NULL,                -- primary server status 0: none, 1: current server, 2: ready to
                                                          failback
  redn_stdby_status   SMALLINT NOT NULL,                -- standby server status 0: none, 1: current server, 2: ready to
                                                          failover
  redn_temp_status    SMALLINT NOT NULL,                -- secondary primary server status 0: none, 1: existing (under
                                                          monitoring),  2: rebooting
  redn_timestamp      DATETIME,                         -- time stamp for status checked
  redn_failure_id     INTEGER,                          -- failed service ID, 0 for system
  redn_return         SMALLINT NULL,                    --return value from command line, script or API
  redn_msg            CHAR(256)                         -- failure information
);
```

CREATE TABLE redundancy_cfg

```
(
  site                      CHAR(32) NOT NULL,                          -- Site Name
  redncfg_id                INTEGER IDENTITY NOT FOR REPLICATION PRIMARY KEY,  -- unique database ID
  redncfg_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  redncfg_type              SMALLINT NOT NULL,                          -- server type, 0 = Primary, 1 = Standby
  redncfg_checksum          INTEGER NOT NULL,                           -- Record checksum
  redncfg_server_name       varchar(32) NOT NULL,                       -- server name
  redncfg_server_ip         CHAR(16) NOT NULL,                          -- server IP address
  redncfg_server_datanet_ip CHAR(16) NOT NULL,                          -- server private IP address
```

```
  redncfg_server_login_name          varchar(32) NOT NULL,                -- server login name
  redncfg_server_login_password      CHAR(16),                            -- server login Pswrd
  redncfg_server_domain              varchar(32),                         -- server domain name
  redncfg_safemode_name              varchar(32),                         -- safe mode server name
  redncfg_safemode_ip                CHAR(16),                            -- safe mode server IP address
  redncfg_safemode_login_name        varchar(32),                         -- safe mode server login name
  redncfg_safemode_login_password    CHAR(16),                            -- safe mode server login pswrd
  redncfg_safemode_datanet_ip        CHAR(16),                            -- safe mode datanet IP address
  redncfg_monitor_interval           SMALLINT,                            -- failure checking frequency in seconds
  redncfg_monitor_timeout            SMALLINT,                            -- failover delay time from failure to start
                                                                             failover procedure
  redncfg_bandwidth_limit            INT,                                 -- network bandwidth limitation between
                                                                             primary server to standby server
  redncfg_database_path              varchar(500) NOT NULL,               -- where the database data file is
  redncfg_failover_public_nic        INT,                                 -- fail over Public NIC number
  redncfg_failover_private_nic       INT                                  -- fail over Private NIC number
);
  CREATE UNIQUE INDEX                redundancy_cfg_site_name ON redundancy_cfg (site, redncfg_server_name);
```

CREATE TABLE redundancy_cfg_save

```
(
  site                               CHAR(32) NOT NULL,                   -- Site Name
  redncfg_id                         INTEGER,                             -- unique database ID
  redncfg_guid                       uniqueidentifier NOT NULL,           -- Global unique record identifier
  redncfg_type                       SMALLINT NOT NULL,                   -- server type, 0 = Primary, 1 = Standby
  redncfg_checksum                   INTEGER NOT NULL,                    -- Record checksum
  redncfg_server_name                varchar(32) NOT NULL,                -- server name
  redncfg_server_ip                  CHAR(16) NOT NULL,                   -- server IP address
  redncfg_server_datanet_ip          CHAR(16) NOT NULL,                   -- server private IP address
  redncfg_server_login_name          varchar(32) NOT NULL,                -- server login name
  redncfg_server_login_password      CHAR(16),                            --  server login Pswrd
  redncfg_server_domain              varchar(32),                         -- server domain name
  redncfg_safemode_name              varchar(32),                         -- safe mode server name
  redncfg_safemode_ip                CHAR(16),                            --  safe mode server IP address
  redncfg_safemode_login_name        varchar(32),                         --  safe mode server login name
  redncfg_safemode_login_password    CHAR(16),                            --  safe mode server login pswrd
  redncfg_safemode_datanet_ip        CHAR(16),                            --  safe mode datanet IP address
  redncfg_monitor_interval           SMALLINT,                            -- failure checking frequency in seconds
  redncfg_monitor_timeout            SMALLINT,                            -- failover delay time from failure to start failover procedure
  redncfg_bandwidth_limit            INT,                                 -- network bandwidth limitation between primary server to
                                                                             standby server
  redncfg_database_path              varchar(500) NOT NULL,               -- where the database data file is
  redncfg_failover_public_nic        INT,                                 -- fail over Public NIC number
  redncfg_failover_private_nic       INT,                                 -- fail over Private NIC number
  upd_guid                           uniqueidentifier NOT NULL,           -- Guid for update operation
  upd_mode                           SMALLINT NOT NULL,                   -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                      DATETIME NOT NULL,                   -- Date / Time of the update
  upd_checksum                       INTEGER NOT NULL,                    -- Checksum
  upd_rowguid                        uniqueidentifier ROWGUIDCOL DEFAULT
                                       newid() UNIQUE NOT NULL,           -- Global unique record identifier
  PRIMARY KEY                        ( redncfg_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE redundancy_isolation_ip

```
(
  site                    CHAR(32) NOT NULL,                              -- Site Name
  rii_id                  INTEGER IDENTITY NOT FOR REPLICATION PRIMARY KEY,
  rii_guid                uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
  rii_checksum            INTEGER NOT NULL,                               -- Record checksum
  rii_ip                  CHAR(16) NOT NULL,                              -- isolation IP address
  rii_desc                CHAR(64),                                       -- description of the IP
);
```

CREATE TABLE redundancy_isolation_ip_save

```
(
  site                CHAR(32) NOT NULL,                                          -- Site Name
  rii_id              INTEGER NOT NULL,
  rii_guid            uniqueidentifier NOT NULL,                                  -- Global unique record identifier
  rii_checksum        INTEGER NOT NULL,                                           -- Record checksum
  rii_ip              CHAR(16) NOT NULL,                                          -- isolation IP address
  rii_desc            CHAR(64),                                                   -- description of the IP
  upd_guid            uniqueidentifier NOT NULL,                                  -- Guid for update operation
  upd_mode            SMALLINT NOT NULL,                                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp       DATETIME NOT NULL,                                          -- Date / Time of the update
  upd_checksum        INTEGER NOT NULL,                                           -- Checksum
  upd_rowguid         uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY         ( rii_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE registration

```
(
  r_reg_key           CHAR(40) NULL,                          -- registration key
  r_install_key       CHAR(24) PRIMARY KEY NOT NULL,          -- installation key
  r_validity_key      CHAR(16) NULL                           -- validity key
);
```

CREATE TABLE remoteserver

```
(
  rs_id                 INTEGER IDENTITY NOT FOR REPLICATION
                          PRIMARY KEY,                      -- unique database ID
  rs_name               CHAR(32) NOT NULL,                  -- remote server name
  rs_partition          INTEGER REFERENCES partition(part_number)
                          NOT FOR REPLICATION,              -- partition ID to which this remote server belongs
  rs_public             SMALLINT NOT NULL,                  -- 1 = remote server is public, 0 = remote server is private
  rs_computer           CHAR(255) NULL,                     -- computer name
  rs_ip_address         CHAR(16) NULL,                      -- TCP IP Address
  rs_port               INTEGER NULL,                       -- TCP IP Port number
  rs_tz_id              INTEGER REFERENCES timezone(tz_id)
                          NOT FOR REPLICATION,              -- time zone ID
  rs_mfg_id             INTEGER REFERENCES msgfiltergroup(mfg_id)
                          NOT FOR REPLICATION,              -- message filter group ID
  rs_queuelength        INTEGER NOT NULL,                   -- queue length 100 to 1000
  rs_queueexpiry        INTEGER NOT NULL,                   -- queue expiry 15 to 900
  rs_sessionlength      INTEGER NOT NULL,                   -- session length 1 to 100
  rs_sessionduration    INTEGER NOT NULL,                   -- session duration 60 to 900
  rs_sessiontimeout     INTEGER NOT NULL,                   -- session end timeout 15 to 900
  rs_remote_site        CHAR(32),                           -- site name of the remote site
  rs_enable_transmit    SMALLINT NOT NULL,                  -- enable tranmission to remote computer
  rs_enable_receive     SMALLINT NOT NULL,                  -- enable receiving information from remote site
  rs_invalid_attempt    INTEGER NOT NULL,                   -- number of consecutive invalid attempts
  site                  CHAR(32) NOT NULL,                  -- Site Name
  rs_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  rs_checksum           INTEGER NOT NULL,                   -- Record checksum
  rs_protocol           SMALLINT NOT NULL,                  -- 0 = P2000 binary, 1 = P2000 XML
);
  CREATE UNIQUE INDEX   rs_site_name ON remoteserver (site, rs_name);
```

CREATE TABLE remoteserver_save

```
(
  rs_id                 INTEGER NOT NULL,                   -- unique database ID
  rs_name               CHAR(32)  NOT NULL,                 -- remote server name
  rs_partition          INTEGER,                            -- partition ID to which this remote server belongs
  rs_public             SMALLINT NOT NULL,                  -- 1 = remote server is public, 0 = remote server is private
  rs_computer           CHAR(255) NULL,                     -- computer name
```

```
  rs_ip_address          CHAR(16)  NULL,                                  -- TCP IP Address
  rs_port                INTEGER NULL,                                  -- TCP IP Port number
  rs_tz_id               INTEGER,                                       -- time zone ID
  rs_mfg_id              INTEGER,                                       -- message filter group ID
  rs_queuelength         INTEGER NOT NULL,                              -- queue length 100 to 1000
  rs_queueexpiry         INTEGER NOT NULL,                              -- queue expiry 15 to 900
  rs_sessionlength       INTEGER NOT NULL,                              -- session length 1 to 100
  rs_sessionduration     INTEGER NOT NULL,                              -- session duration 60 to 900
  rs_sessiontimeout      INTEGER NOT NULL,                              -- session end timeout 15 to 900
  rs_remote_site         CHAR(32),                                      -- site name of the remote site
  rs_enable_transmit     SMALLINT NOT NULL,                             -- enable tranmission to remote computer
  rs_enable_receive      SMALLINT NOT NULL,                             -- enable receiving information from remote site
  rs_invalid_attempt     INTEGER  NOT NULL,                             -- number of consecutive invalid attempts
  site                   CHAR(32) NOT NULL,                             -- Site Name
  rs_guid                uniqueidentifier NOT NULL,                     -- Global unique record identifier
  rs_checksum            INTEGER NOT NULL,                              -- Record checksum
  upd_guid               uniqueidentifier NOT NULL,                     -- Guid for update operation
  upd_mode               SMALLINT NOT NULL,                             -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp          DATETIME NOT NULL,                             -- Date / Time of the update
  upd_checksum           INTEGER NOT NULL,                              -- Checksum
  upd_rowguid            uniqueidentifier ROWGUIDCOL DEFAULT newid()
                           UNIQUE NOT NULL,                             -- Global unique record identifier
  rs_protocol            SMALLINT NOT NULL,                             -- 0 = P2000 binary, 1 = P2000 XML
  PRIMARY KEY            (rs_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE report

```
(
  report_partition       INTEGER REFERENCES partition(part_number)
                           NOT FOR REPLICATION,                         -- partition ID to which this report belongs
  report_public          SMALLINT,                                      -- 1 = public, 0 = private
  report_id              INTEGER IDENTITY NOT FOR REPLICATION
                           PRIMARY KEY,                                 -- unique report ID
  report_name            CHAR(128) NOT NULL,                            -- report name
  report_user            SMALLINT,                                      -- 1 = user report
  report_lang_id         CHAR(4) NOT NULL,                              -- report lang id (hex) e.g. spanish = 040a, english 0409
  report_index           SMALLINT,                                      -- report index id
  site                   CHAR(32) NOT NULL,                             -- Site Name
  report_guid            uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
  report_checksum        INTEGER NOT NULL,                              -- Record checksum
  report_hide            SMALLINT NOT NULL,                             -- 1 = report is hidden
);
  CREATE UNIQUE INDEX    rpt_part_name ON report (site,  report_partition, report_name, report_user, report_lang_id );
  CREATE INDEX           report_public ON report ( report_public );
```

CREATE TABLE reportdata

```
(
  report_id              INTEGER REFERENCES report(report_id) NOT FOR REPLICATION,   -- report ID
  report_data            IMAGE,                                                      -- report binary data
  site                   CHAR(32) NOT NULL,                                          -- Site Name
  report_guid            uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                -- Global unique record identifier
  report_checksum        INTEGER NOT NULL,                                           -- Record checksum
  PRIMARY KEY            (report_id)
);
```

CREATE TABLE reportdata_save

```
(
  report_id              INTEGER NOT NULL,                                           -- report ID
  report_data            IMAGE,                                                      -- report binary data
  site                   CHAR(32) NOT NULL,                                          -- Site Name
  report_guid            uniqueidentifier NOT NULL,                                  -- Global unique record identifier
```

```
  report_checksum       INTEGER NOT NULL,                                          -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                                 -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                         -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                          -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY           (report_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE reportpartition

```
(
  rp_id                 uniqueidentifier NOT NULL,                 -- Identifier for report instance
  rp_part_name          CHAR(32) NOT NULL,                         -- Partition to use for this report
  PRIMARY KEY           ( rp_id, rp_part_name)
);
```

CREATE TABLE report_save

```
(
  report_partition      INTEGER,                                   -- partition ID to which this report belongs
  report_public         SMALLINT,                                  -- 1 = public, 0 = private
  report_id             INTEGER  NOT NULL,                         -- unique report ID
  report_name           CHAR(128) NOT NULL,                        -- report name
  report_user           SMALLINT,                                  -- 1 = user report
  report_lang_id        CHAR(4) NOT NULL,                          -- report lang id (hex) e.g. spanish = 040a, english 0409
  report_index          SMALLINT,                                  -- report index id
  site                  CHAR(32) NOT NULL,                         -- Site Name
  report_guid           uniqueidentifier NOT NULL,                 -- Global unique record identifier
  report_checksum       INTEGER NOT NULL,                          -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                 -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                         -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                          -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                         -- Global unique record identifier
  report_hide           SMALLINT,                                  -- 1 = report is hidden
  PRIMARY KEY           (report_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE requestapprovers

```
(
  ra_guid               uniqueidentifier DEFAULT newid() ROWGUIDCOL UNIQUE NOT NULL,
  ra_partition          INTEGER REFERENCES partition(part_number) NOT FOR REPLICATION,
  ra_order              INTEGER NOT NULL,
  ra_sender             CHAR(64) NOT NULL,
  ra_senderdetail       VARCHAR(64) NOT NULL,
  ra_operation          INT NOT NULL,
  ra_company            INTEGER NULL REFERENCES company(company_id) NOT FOR REPLICATION,
  ra_department         INTEGER NULL REFERENCES dept(dept_id) NOT FOR REPLICATION,
  site                  CHAR(32) NOT NULL,
  ra_checksum           INTEGER NOT NULL,
  PRIMARY KEY           ( site, ra_partition, ra_order )
);
```

CREATE TABLE requestapprovers_details

```
(
  rad_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                -- Global unique record identifier
  rad_ra_guid           uniqueidentifier REFERENCES requestapprovers(ra_guid) NOT FOR REPLICATION,
  rad_cardholder        INTEGER REFERENCES cardholder(c_id) NOT FOR REPLICATION,
  rad_active            INTEGER NOT NULL,
  rad_level             INTEGER NOT NULL,
```

```
  site                CHAR(32) NOT NULL,                                          -- Site Name
  rad_checksum        INTEGER NOT NULL,                                           -- Record checksum
  PRIMARY KEY         ( rad_ra_guid, rad_cardholder, rad_level )
);
```

CREATE TABLE requestapprovers_details_save

```
(
  rad_guid-           NOT NULL,                                                   -- Global unique record identifier
  rad_ra_guid         uniqueidentifier,
  rad_cardholder      INTEGER,
  rad_active          INTEGER NOT NULL,
  rad_level           INTEGER NOT NULL,
  site                CHAR(32) NOT NULL,                                          -- Site Name
  rad_checksum        INTEGER NOT NULL,                                           -- Record checksum
  upd_guid            uniqueidentifier NOT NULL,                                  -- Guid for update operation
  upd_mode            SMALLINT NOT NULL,                                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp       DATETIME NOT NULL,                                          -- Date / Time of the update
  upd_checksum        INTEGER NOT NULL,                                           -- Checksum
  upd_rowguid         uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY         ( rad_ra_guid, rad_cardholder, upd_guid )
);
```

CREATE TABLE requestapprovers_save

```
(
  ra_guid             uniqueidentifier NOT NULL,
  ra_partition        INTEGER NOT NULL,
  ra_order            INTEGER NOT NULL,
  ra_sender           CHAR(64) NOT NULL,
  ra_senderdetail     VARCHAR(64) NOT NULL,
  ra_operation        INT NOT NULL,
  ra_company          INTEGER NULL,
  ra_department       INTEGER NULL,
  site                CHAR(32) NOT NULL,
  ra_checksum         INTEGER NOT NULL,
  upd_guid            uniqueidentifier NOT NULL,                                  -- Guid for update operation
  upd_mode            SMALLINT NOT NULL,                                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp       DATETIME NOT NULL,                                          -- Date / Time of the update
  upd_checksum        INTEGER NOT NULL,                                           -- Checksum
  upd_rowguid         uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY         ( ra_partition, ra_order, upd_guid )
);
```

CREATE TABLE requestconfig

```
(
  site                CHAR(32) NOT NULL,                                          -- Site Name
  rc_guid             uniqueidentifier DEFAULT newid()
                        ROWGUIDCOL UNIQUE NOT NULL,                               -- Global unique record identifier
  rc_sender           VARCHAR(64) NOT NULL,                                       -- sender of application
  rc_senderdetail     VARCHAR(64) NOT NULL,                                       -- sender details of application
  rc_approval_levels  INTEGER NOT NULL,                                           -- number of approvals for this application
  rc_process_type     INTEGER NOT NULL,                                           -- 0 = manual processing, 1 = auto processing
  rc_checksum         INTEGER NOT NULL,                                           -- Record checksum
  PRIMARY KEY         ( site, rc_sender, rc_senderdetail )
);
```

CREATE TABLE requestconfig_save

```
(
  site                CHAR(32) NOT NULL,                                          -- Site Name
  rc_guid             uniqueidentifier NOT NULL,                                  -- Global unique record identifier
  rc_sender           VARCHAR(64) NOT NULL,                                       -- sender of application
```

```
   rc_senderdetail          VARCHAR(64) NOT NULL,                          -- sender details of application
   rc_approval_levels       INTEGER NOT NULL,                              -- number of approvals for this application
   rc_process_type          INTEGER NOT NULL,                              -- 0 = manual processing, 1 = auto processing
   rc_checksum              INTEGER NOT NULL,                              -- Record checksum
   upd_guid                 uniqueidentifier NOT NULL,                     -- Guid for update operation
   upd_mode                 SMALLINT NOT NULL,                             -- Update mode (0 - delete, 1 - edit, 2 - new)
   upd_timestamp            DATETIME NOT NULL,                             -- Date / Time of the update
   upd_checksum             INTEGER NOT NULL,                              -- Checksum
   upd_rowguid              uniqueidentifier ROWGUIDCOL DEFAULT newid()
                               UNIQUE NOT NULL,                            -- Global unique record identifier
   PRIMARY KEY              ( site, rc_sender, rc_senderdetail, upd_guid )
);
```

CREATE TABLE request_queue

```
(
   rq_id                    INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,   -- unique database ID,
   rq_sender                VARCHAR(64) NULL,
   rq_senderdetail          VARCHAR(64) NULL,
   rq_sourceip              VARCHAR(32) NULL,
   rq_operation             TINYINT NOT NULL,
   rq_createtime            datetime NOT NULL,
   rq_modificationtime      datetime NOT NULL,
   rq_revised               BIT NOT NULL,
   rq_processstatus         TINYINT NOT NULL,
   rq_integrity             BIT NOT NULL,
   rq_rfirstname            VARCHAR(25) NULL,
   rq_rmiddlename           VARCHAR(25) NULL,
   rq_rlastname             VARCHAR(25) NULL,
   rq_rsitename             VARCHAR(32) NULL,
   rq_rpartition            VARCHAR(32) NULL,
   rq_rcompany              VARCHAR(32) NULL,
   rq_requestnotes          VARCHAR(255) NULL,
   rq_entityid              VARCHAR(25) NULL,
   rq_type                  TINYINT NOT NULL,
   rq_sitename              VARCHAR(32) NULL,
   rq_partition             VARCHAR(32) NULL,
   rq_firstname             VARCHAR(25) NOT NULL,
   rq_middlename            VARCHAR(25) NULL,
   rq_lastname              VARCHAR(25) NOT NULL,
   rq_company               VARCHAR(32) NULL,
   rq_starttime             datetime NULL,
   rq_expiretime            datetime NULL,
   rq_sfirstname            VARCHAR(25) NULL,
   rq_smiddlename           VARCHAR(25) NULL,
   rq_slastname             VARCHAR(25) NULL,
   rq_ssitename             VARCHAR(32) NULL,
   rq_spartition            VARCHAR(32) NULL,
   rq_scompany              VARCHAR(32) NULL,
   rq_orgxmlblock           TEXT NOT NULL,
   rq_rltxmlblock           TEXT NULL,
   rq_rdepartment           VARCHAR(32) NULL,
   rq_approver_guid         uniqueidentifier NULL,
   rowtimestamp             timestamp,
   rq_location              VARCHAR(255) NULL,
   rq_timeofarrival         datetime NULL,
   PRIMARY KEY              ( rq_id )
);
   CREATE INDEX             rq_processstatus ON request_queue ( rq_processstatus );
   CREATE INDEX             rq_name ON request_queue ( rq_lastname, rq_firstname, rq_middlename );
   CREATE INDEX             rq_sname ON request_queue ( rq_slastname, rq_sfirstname, rq_smiddlename );
   CREATE INDEX             rq_rname ON request_queue ( rq_rlastname, rq_rfirstname, rq_rmiddlename );
   CREATE INDEX             rq_company ON request_queue ( rq_company );
   CREATE INDEX             rq_scompany ON request_queue ( rq_scompany );
```

```
CREATE INDEX        rq_rcompany ON request_queue ( rq_rcompany );
CREATE INDEX        rq_starttime ON request_queue ( rq_starttime );
CREATE INDEX        rq_expiretime ON request_queue ( rq_expiretime );
CREATE INDEX        rq_createtime ON request_queue ( rq_createtime );
CREATE INDEX        rq_approver_guid ON request_queue ( rq_approver_guid );
```

CREATE TABLE request_queue_hist

```
(
  rq_guid               uniqueidentifier DEFAULT newid() ROWGUIDCOL NOT NULL,        -- unique id for this record
  rq_sender             VARCHAR(64) NULL,
  rq_senderdetail       VARCHAR(64) NULL,
  rq_sourceip           VARCHAR(32) NULL,
  rq_operation          TINYINT NOT NULL,
  rq_createtime         datetime NOT NULL,
  rq_modificationtime   datetime NOT NULL,
  rq_revised            BIT NOT NULL,
  rq_processstatus      TINYINT NOT NULL,
  rq_integrity          BIT NOT NULL,
  rq_rfirstname         VARCHAR(25) NULL,
  rq_rmiddlename        VARCHAR(25) NULL,
  rq_rlastname          VARCHAR(25) NULL,
  rq_rsitename          VARCHAR(32) NULL,
  rq_rpartition         VARCHAR(32) NULL,
  rq_rcompany           VARCHAR(32) NULL,
  rq_requestnotes       VARCHAR(255) NULL,
  rq_entityid           VARCHAR(25) NULL,
  rq_type               TINYINT NOT NULL,
  rq_sitename           VARCHAR(32) NULL,
  rq_partition          VARCHAR(32) NULL,
  rq_firstname          VARCHAR(25) NOT NULL,
  rq_middlename         VARCHAR(25) NULL,
  rq_lastname           VARCHAR(25) NOT NULL,
  rq_company            VARCHAR(32) NULL,
  rq_starttime          datetime NULL,
  rq_expiretime         datetime NULL,
  rq_sfirstname         VARCHAR(25) NULL,
  rq_smiddlename        VARCHAR(25) NULL,
  rq_slastname          VARCHAR(25) NULL,
  rq_ssitename          VARCHAR(32) NULL,
  rq_spartition         VARCHAR(32) NULL,
  rq_scompany           VARCHAR(32) NULL,
  rq_orgxmlblock        TEXT NOT NULL,
  rq_rltxmlblock        TEXT NULL,
  rq_rdepartment        VARCHAR(32) NULL,
  rq_approver_guid      uniqueidentifier NULL,
  rq_location           VARCHAR(255) NULL,
  rq_timeofarrival      datetime NULL,
  PRIMARY KEY           NONCLUSTERED ( rq_guid )
);
  CREATE CLUSTERED INDEX    rq_createtime ON request_queue_hist ( rq_createtime );
  CREATE INDEX              rq_processstatus ON request_queue_hist ( rq_processstatus );
  CREATE INDEX              rq_name ON request_queue_hist ( rq_lastname, rq_firstname, rq_middlename );
  CREATE INDEX              rq_sname ON request_queue_hist ( rq_slastname, rq_sfirstname, rq_smiddlename );
  CREATE INDEX              rq_rname ON request_queue_hist ( rq_rlastname, rq_rfirstname, rq_rmiddlename );
  CREATE INDEX              rq_company ON request_queue_hist ( rq_company );
  CREATE INDEX              rq_scompany ON request_queue_hist ( rq_scompany );
  CREATE INDEX              rq_rcompany ON request_queue_hist ( rq_rcompany );
  CREATE INDEX              rq_starttime ON request_queue_hist ( rq_starttime );
  CREATE INDEX              rq_expiretime ON request_queue_hist ( rq_expiretime );
```

CREATE TABLE requiredfields

```
(
  rf_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
  rf_field         int NOT NULL,                                  -- enum value of field
  site             CHAR(32) NOT NULL,                             -- Site Name
  rf_checksum      INTEGER NOT NULL,                              -- Record checksum
  PRIMARY KEY      ( rf_field, site )
);
```

CREATE TABLE requiredfields_save

```
(
  rf_guid          uniqueidentifier NOT NULL,                     -- Global unique record identifier
  rf_field         int NOT NULL,                                  -- enum value of field
  site             CHAR(32) NOT NULL,                             -- Site Name
  rf_checksum      INTEGER NOT NULL,                              -- Record checksum
  upd_guid         uniqueidentifier NOT NULL,                     -- Guid for update operation
  upd_mode         SMALLINT NOT NULL,                             -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp    DATETIME NOT NULL,                             -- Date / Time of the update
  upd_checksum     INTEGER NOT NULL,                              -- Checksum
  upd_rowguid      uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,  -- Global unique record identifier
  PRIMARY KEY      ( rf_field, site, upd_mode desc, upd_guid  )
);
```

CREATE TABLE rtlroute

```
(
  rtlroute_id      INTEGER IDENTITY NOT FOR REPLICATION PRIMARY KEY,  -- unique rtlroute ID
  rtlroute_data    VARBINARY(7172) NOT NULL,                          -- up to 7172 bytes of data
);
```

CREATE TABLE scheduled_actions

```
(
  sa_id            INTEGER IDENTITY NOT FOR REPLICATION PRIMARY KEY,  -- unique scheduled action ID
  sa_execute_at    DATETIME NOT NULL,                                 -- time to execute at
  sa_action_id     INTEGER NOT NULL REFERENCES event_action(eva_id)
                     NOT FOR REPLICATION,                             -- event action ID to execute
  sa_event_id      INTEGER REFERENCES event(ev_id) NOT FOR REPLICATION,  -- event ID for this action
);
  CREATE INDEX     sa_execute_at ON scheduled_actions ( sa_execute_at );
```

CREATE TABLE SecLevelRanges

```
(
  slr_min_green    INTEGER NOT NULL,
  slr_max_green    INTEGER NOT NULL,
  slr_setto_green  INTEGER NOT NULL,
  slr_min_blue     INTEGER NOT NULL,
  slr_max_blue     INTEGER NOT NULL,
  slr_setto_blue   INTEGER NOT NULL,
  slr_min_yellow   INTEGER NOT NULL,
  slr_max_yellow   INTEGER NOT NULL,
  slr_setto_yellow INTEGER NOT NULL,
  slr_min_orange   INTEGER NOT NULL,
  slr_max_orange   INTEGER NOT NULL,
  slr_setto_orange INTEGER NOT NULL,
  slr_min_red      INTEGER NOT NULL,
  slr_max_red      INTEGER NOT NULL,
  slr_setto_red    INTEGER NOT NULL,
  slr_enable_sl    SMALLINT NOT NULL,
  slr_id           INTEGER IDENTITY NOT FOR REPLICATION NOT NULL,
  site             CHAR(32) NOT NULL,                             -- Site Name
  slr_guid         uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
```

```
  slr_checksum          INTEGER NOT NULL,                                    -- Record checksum
  PRIMARY KEY           ( slr_id )
);
```

CREATE TABLE SecLevelRanges_save

```
(
  slr_min_green         INTEGER NOT NULL,
  slr_max_green         INTEGER NOT NULL,
  slr_setto_green       INTEGER NOT NULL,
  slr_min_blue          INTEGER NOT NULL,
  slr_max_blue          INTEGER NOT NULL,
  slr_setto_blue        INTEGER NOT NULL,
  slr_min_yellow        INTEGER NOT NULL,
  slr_max_yellow        INTEGER NOT NULL,
  slr_setto_yellow      INTEGER NOT NULL,
  slr_min_orange        INTEGER NOT NULL,
  slr_max_orange        INTEGER NOT NULL,
  slr_setto_orange      INTEGER NOT NULL,
  slr_min_red           INTEGER NOT NULL,
  slr_max_red           INTEGER NOT NULL,
  slr_setto_red         INTEGER NOT NULL,
  slr_enable_sl         SMALLINT NOT NULL,
  slr_id                INTEGER NOT NULL,
  site                  CHAR(32) NOT NULL,                                   -- Site Name
  slr_guid              uniqueidentifier NOT NULL,                           -- Global unique record identifier
  slr_checksum          INTEGER NOT NULL,                                    -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                           -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                   -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                   -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                    -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY           ( slr_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE service_stat

```
(
  svrstat_service_id    INTEGER REFERENCES svr_startup_cnfg(svr_id) NOT FOR REPLICATION,  -- service ID
  svrstat_session_guid  uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,         -- session guid for this connection
  site                  CHAR(32) NOT NULL,                                   -- Site Name
  svrstat_guid          uniqueidentifier PRIMARY KEY,                        -- Record guid for this connection
  svrstat_checksum      INTEGER NOT NULL,                                    -- Record checksum
);
```

CREATE TABLE siadevice

```
(
  site                  CHAR(32) NOT NULL,                                   -- Site Name
  sia_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,         -- Global unique record identifier
  sia_partition         INTEGER REFERENCES partition(part_number)
                          NOT FOR REPLICATION,                               -- partition ID to which this device belongs
  sia_public            SMALLINT,                                            -- 1 = public, 0 = private
  sia_name              CHAR(32) NOT NULL,                                   -- name of device
  sia_enable            SMALLINT NOT NULL,                                   -- 1 = enable, 0 = disabled
  sia_port              SMALLINT NOT NULL,                                   -- COM port number
  sia_baud_rate         INTEGER NOT NULL,                                    -- COM port baud rate
  sia_data_bits         SMALLINT NOT NULL,                                   -- COM port data bits
  sia_parity            SMALLINT NOT NULL,                                   -- COM port parity
  sia_stop_bits         SMALLINT NOT NULL,                                   -- COM port stop bits
  sia_checksum          INTEGER NOT NULL,                                    -- Record checksum
  sia_disp_alarm        SMALLINT NOT NULL,                                   -- 1 = display alarm in alarm monitor
  PRIMARY KEY           (site, sia_port)
);
  CREATE UNIQUE INDEX   sia_site_name ON siadevice (site, sia_name);
```

CREATE TABLE siadevice_save

```
(
  site              CHAR(32) NOT NULL,                                          -- Site Name
  sia_guid          uniqueidentifier NOT NULL,                                  -- Global unique record identifier
  sia_partition     INTEGER,                                                    -- partition ID to which this device belongs
  sia_public        SMALLINT,                                                   -- 1 = public, 0 = private
  sia_name          CHAR(32) NOT NULL,                                          -- name of device
  sia_enable        SMALLINT NOT NULL,                                          -- 1 = enable, 0 = disabled
  sia_port          SMALLINT NOT NULL,                                          -- COM port number
  sia_baud_rate     INTEGER NOT NULL,                                           -- COM port baud rate
  sia_data_bits     SMALLINT NOT NULL,                                          -- COM port data bits
  sia_parity        SMALLINT NOT NULL,                                          -- COM port parity
  sia_stop_bits     SMALLINT NOT NULL,                                          -- COM port stop bits
  sia_checksum      INTEGER NOT NULL,                                           -- Record checksum
  upd_guid          uniqueidentifier NOT NULL,                                  -- Guid for update operation
  upd_mode          SMALLINT NOT NULL,                                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp     DATETIME NOT NULL,                                          -- Date / Time of the update
  upd_checksum      INTEGER NOT NULL,                                           -- Checksum
  upd_rowguid       uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  sia_disp_alarm    SMALLINT,                                                   -- 1 = display alarm in alarm monitor
  PRIMARY KEY       (site, sia_port, upd_guid)
);
```

CREATE TABLE siddnld

```
(
  sid_id            INTEGER IDENTITY NOT FOR REPLICATION
                      PRIMARY KEY,                                              -- unique siddnld ID
  sid_dnld_type     SMALLINT,                                                   -- 1 = CK720 network panel, 2 = serial interface (legacy) panel
  sid_panel_id      INTEGER,                                                    -- panel ID to download to
  sid_timestamp     DATETIME,                                                   -- time of dnld data
  sid_nak_count     SMALLINT,                                                   -- number of NAK replies from panel
  sid_count         SMALLINT,                                                   -- data count in bytes
  sid_data_0        INTEGER NULL,                                               -- 4 bytes of data
  sid_data_1        INTEGER NULL,                                               -- 4 bytes of data
  sid_data_2        INTEGER NULL,                                               -- 4 bytes of data
  sid_data_3        INTEGER NULL,                                               -- 4 bytes of data
  sid_data_4        INTEGER NULL,                                               -- 4 bytes of data
  sid_data_5        INTEGER NULL,                                               -- 4 bytes of data
  sid_data_6        INTEGER NULL,                                               -- 4 bytes of data
  sid_data_7        INTEGER NULL,                                               -- 4 bytes of data
  sid_data_8        INTEGER NULL,                                               -- 4 bytes of data
  sid_data_9        INTEGER NULL,                                               -- 4 bytes of data
  sid_data_10       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_11       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_12       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_13       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_14       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_15       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_16       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_17       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_18       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_19       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_20       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_21       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_22       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_23       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_24       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_25       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_26       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_27       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_28       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_29       INTEGER NULL,                                               -- 4 bytes of data
  sid_data_30       INTEGER NULL,                                               -- 4 bytes of data
```

```
sid_data_31     INTEGER NULL,     -- 4 bytes of data
sid_data_32     INTEGER NULL,     -- 4 bytes of data
sid_data_33     INTEGER NULL,     -- 4 bytes of data
sid_data_34     INTEGER NULL,     -- 4 bytes of data
sid_data_35     INTEGER NULL,     -- 4 bytes of data
sid_data_36     INTEGER NULL,     -- 4 bytes of data
sid_data_37     INTEGER NULL,     -- 4 bytes of data
sid_data_38     INTEGER NULL,     -- 4 bytes of data
sid_data_39     INTEGER NULL,     -- 4 bytes of data
sid_data_40     INTEGER NULL,     -- 4 bytes of data
sid_data_41     INTEGER NULL,     -- 4 bytes of data
sid_data_42     INTEGER NULL,     -- 4 bytes of data
sid_data_43     INTEGER NULL,     -- 4 bytes of data
sid_data_44     INTEGER NULL,     -- 4 bytes of data
sid_data_45     INTEGER NULL,     -- 4 bytes of data
sid_data_46     INTEGER NULL,     -- 4 bytes of data
sid_data_47     INTEGER NULL,     -- 4 bytes of data
sid_data_48     INTEGER NULL,     -- 4 bytes of data
sid_data_49     INTEGER NULL,     -- 4 bytes of data
sid_data_50     INTEGER NULL,     -- 4 bytes of data
sid_data_51     INTEGER NULL,     -- 4 bytes of data
sid_data_52     INTEGER NULL,     -- 4 bytes of data
sid_data_53     INTEGER NULL,     -- 4 bytes of data
sid_data_54     INTEGER NULL,     -- 4 bytes of data
sid_data_55     INTEGER NULL,     -- 4 bytes of data
sid_data_56     INTEGER NULL,     -- 4 bytes of data
sid_data_57     INTEGER NULL,     -- 4 bytes of data
sid_data_58     INTEGER NULL,     -- 4 bytes of data
sid_data_59     INTEGER NULL,     -- 4 bytes of data
sid_data_60     INTEGER NULL,     -- 4 bytes of data
sid_data_61     INTEGER NULL,     -- 4 bytes of data
sid_data_62     INTEGER NULL,     -- 4 bytes of data
sid_data_63     INTEGER NULL,     -- 4 bytes of data
sid_data_64     INTEGER NULL,     -- 4 bytes of data
sid_data_65     INTEGER NULL,     -- 4 bytes of data
sid_data_66     INTEGER NULL,     -- 4 bytes of data
sid_data_67     INTEGER NULL,     -- 4 bytes of data
sid_data_68     INTEGER NULL,     -- 4 bytes of data
sid_data_69     INTEGER NULL,     -- 4 bytes of data
sid_data_70     INTEGER NULL,     -- 4 bytes of data
sid_data_71     INTEGER NULL,     -- 4 bytes of data
sid_data_72     INTEGER NULL,     -- 4 bytes of data
sid_data_73     INTEGER NULL,     -- 4 bytes of data
sid_data_74     INTEGER NULL,     -- 4 bytes of data
sid_data_75     INTEGER NULL,     -- 4 bytes of data
sid_data_76     INTEGER NULL,     -- 4 bytes of data
sid_data_77     INTEGER NULL,     -- 4 bytes of data
sid_data_78     INTEGER NULL,     -- 4 bytes of data
sid_data_79     INTEGER NULL,     -- 4 bytes of data
sid_data_80     INTEGER NULL,     -- 4 bytes of data
sid_data_81     INTEGER NULL,     -- 4 bytes of data
sid_data_82     INTEGER NULL,     -- 4 bytes of data
sid_data_83     INTEGER NULL,     -- 4 bytes of data
sid_data_84     INTEGER NULL,     -- 4 bytes of data
sid_data_85     INTEGER NULL,     -- 4 bytes of data
sid_data_86     INTEGER NULL,     -- 4 bytes of data
sid_data_87     INTEGER NULL,     -- 4 bytes of data
sid_data_88     INTEGER NULL,     -- 4 bytes of data
sid_data_89     INTEGER NULL,     -- 4 bytes of data
sid_data_90     INTEGER NULL,     -- 4 bytes of data
sid_data_91     INTEGER NULL,     -- 4 bytes of data
sid_data_92     INTEGER NULL,     -- 4 bytes of data
sid_data_93     INTEGER NULL,     -- 4 bytes of data
sid_data_94     INTEGER NULL,     -- 4 bytes of data
sid_data_95     INTEGER NULL,     -- 4 bytes of data
```

```
  sid_data_96           INTEGER NULL,                    -- 4 bytes of data
  sid_data_97           INTEGER NULL,                    -- 4 bytes of data
  sid_data_98           INTEGER NULL,                    -- 4 bytes of data
  sid_data_99           INTEGER NULL,                    -- 4 bytes of data
  sid_data_100          INTEGER NULL,                    -- 4 bytes of data
  sid_data_101          INTEGER NULL,                    -- 4 bytes of data
  sid_data_102          INTEGER NULL,                    -- 4 bytes of data
  sid_data_103          INTEGER NULL,                    -- 4 bytes of data
  sid_data_104          INTEGER NULL,                    -- 4 bytes of data
  sid_data_105          INTEGER NULL,                    -- 4 bytes of data
  sid_data_106          INTEGER NULL,                    -- 4 bytes of data
  sid_data_107          INTEGER NULL,                    -- 4 bytes of data
  sid_data_108          INTEGER NULL,                    -- 4 bytes of data
  sid_data_109          INTEGER NULL,                    -- 4 bytes of data
  sid_data_110          INTEGER NULL,                    -- 4 bytes of data
  sid_data_111          INTEGER NULL,                    -- 4 bytes of data
  sid_data_112          INTEGER NULL,                    -- 4 bytes of data
  sid_data_113          INTEGER NULL,                    -- 4 bytes of data
  sid_data_114          INTEGER NULL,                    -- 4 bytes of data
  sid_data_115          INTEGER NULL,                    -- 4 bytes of data
  sid_data_116          INTEGER NULL,                    -- 4 bytes of data
  sid_data_117          INTEGER NULL,                    -- 4 bytes of data
  sid_data_118          INTEGER NULL,                    -- 4 bytes of data
  sid_data_119          INTEGER NULL                     -- 4 bytes of data
);
  CREATE INDEX          sid_panel_id ON siddnld ( sid_panel_id );
  CREATE INDEX          sid_timestamp ON siddnld ( sid_timestamp );
```

CREATE TABLE siddnldpriority

```
(
  sid_id                INTEGER IDENTITY NOT FOR REPLICATION
                          PRIMARY KEY,                   -- unique siddnld ID
  sid_dnld_type         SMALLINT,                        -- 1 = CK720 network panel, 2 = serial interface (legacy) panel
  sid_panel_id          INTEGER,                         -- panel ID to download to
  sid_timestamp         DATETIME,                        -- time of dnld data
  sid_nak_count         SMALLINT,                        -- number of NAK replies from panel
  sid_count             SMALLINT,                        -- data count in bytes
  sid_data_0            INTEGER NULL,                    -- 4 bytes of data
  sid_data_1            INTEGER NULL,                    -- 4 bytes of data
  sid_data_2            INTEGER NULL,                    -- 4 bytes of data
  sid_data_3            INTEGER NULL,                    -- 4 bytes of data
  sid_data_4            INTEGER NULL,                    -- 4 bytes of data
  sid_data_5            INTEGER NULL,                    -- 4 bytes of data
  sid_data_6            INTEGER NULL,                    -- 4 bytes of data
  sid_data_7            INTEGER NULL,                    -- 4 bytes of data
  sid_data_8            INTEGER NULL,                    -- 4 bytes of data
  sid_data_9            INTEGER NULL,                    -- 4 bytes of data
  sid_data_10           INTEGER NULL,                    -- 4 bytes of data
  sid_data_11           INTEGER NULL,                    -- 4 bytes of data
  sid_data_12           INTEGER NULL,                    -- 4 bytes of data
  sid_data_13           INTEGER NULL,                    -- 4 bytes of data
  sid_data_14           INTEGER NULL,                    -- 4 bytes of data
  sid_data_15           INTEGER NULL,                    -- 4 bytes of data
  sid_data_16           INTEGER NULL,                    -- 4 bytes of data
  sid_data_17           INTEGER NULL,                    -- 4 bytes of data
  sid_data_18           INTEGER NULL,                    -- 4 bytes of data
  sid_data_19           INTEGER NULL,                    -- 4 bytes of data
  sid_data_20           INTEGER NULL,                    -- 4 bytes of data
  sid_data_21           INTEGER NULL,                    -- 4 bytes of data
  sid_data_22           INTEGER NULL,                    -- 4 bytes of data
  sid_data_23           INTEGER NULL,                    -- 4 bytes of data
  sid_data_24           INTEGER NULL,                    -- 4 bytes of data
  sid_data_25           INTEGER NULL,                    -- 4 bytes of data
```

```
sid_data_26     INTEGER NULL,     -- 4 bytes of data
sid_data_27     INTEGER NULL,     -- 4 bytes of data
sid_data_28     INTEGER NULL,     -- 4 bytes of data
sid_data_29     INTEGER NULL,     -- 4 bytes of data
sid_data_30     INTEGER NULL,     -- 4 bytes of data
sid_data_31     INTEGER NULL,     -- 4 bytes of data
sid_data_32     INTEGER NULL,     -- 4 bytes of data
sid_data_33     INTEGER NULL,     -- 4 bytes of data
sid_data_34     INTEGER NULL,     -- 4 bytes of data
sid_data_35     INTEGER NULL,     -- 4 bytes of data
sid_data_36     INTEGER NULL,     -- 4 bytes of data
sid_data_37     INTEGER NULL,     -- 4 bytes of data
sid_data_38     INTEGER NULL,     -- 4 bytes of data
sid_data_39     INTEGER NULL,     -- 4 bytes of data
sid_data_40     INTEGER NULL,     -- 4 bytes of data
sid_data_41     INTEGER NULL,     -- 4 bytes of data
sid_data_42     INTEGER NULL,     -- 4 bytes of data
sid_data_43     INTEGER NULL,     -- 4 bytes of data
sid_data_44     INTEGER NULL,     -- 4 bytes of data
sid_data_45     INTEGER NULL,     -- 4 bytes of data
sid_data_46     INTEGER NULL,     -- 4 bytes of data
sid_data_47     INTEGER NULL,     -- 4 bytes of data
sid_data_48     INTEGER NULL,     -- 4 bytes of data
sid_data_49     INTEGER NULL,     -- 4 bytes of data
sid_data_50     INTEGER NULL,     -- 4 bytes of data
sid_data_51     INTEGER NULL,     -- 4 bytes of data
sid_data_52     INTEGER NULL,     -- 4 bytes of data
sid_data_53     INTEGER NULL,     -- 4 bytes of data
sid_data_54     INTEGER NULL,     -- 4 bytes of data
sid_data_55     INTEGER NULL,     -- 4 bytes of data
sid_data_56     INTEGER NULL,     -- 4 bytes of data
sid_data_57     INTEGER NULL,     -- 4 bytes of data
sid_data_58     INTEGER NULL,     -- 4 bytes of data
sid_data_59     INTEGER NULL,     -- 4 bytes of data
sid_data_60     INTEGER NULL,     -- 4 bytes of data
sid_data_61     INTEGER NULL,     -- 4 bytes of data
sid_data_62     INTEGER NULL,     -- 4 bytes of data
sid_data_63     INTEGER NULL,     -- 4 bytes of data
sid_data_64     INTEGER NULL,     -- 4 bytes of data
sid_data_65     INTEGER NULL,     -- 4 bytes of data
sid_data_66     INTEGER NULL,     -- 4 bytes of data
sid_data_67     INTEGER NULL,     -- 4 bytes of data
sid_data_68     INTEGER NULL,     -- 4 bytes of data
sid_data_69     INTEGER NULL,     -- 4 bytes of data
sid_data_70     INTEGER NULL,     -- 4 bytes of data
sid_data_71     INTEGER NULL,     -- 4 bytes of data
sid_data_72     INTEGER NULL,     -- 4 bytes of data
sid_data_73     INTEGER NULL,     -- 4 bytes of data
sid_data_74     INTEGER NULL,     -- 4 bytes of data
sid_data_75     INTEGER NULL,     -- 4 bytes of data
sid_data_76     INTEGER NULL,     -- 4 bytes of data
sid_data_77     INTEGER NULL,     -- 4 bytes of data
sid_data_78     INTEGER NULL,     -- 4 bytes of data
sid_data_79     INTEGER NULL,     -- 4 bytes of data
sid_data_80     INTEGER NULL,     -- 4 bytes of data
sid_data_81     INTEGER NULL,     -- 4 bytes of data
sid_data_82     INTEGER NULL,     -- 4 bytes of data
sid_data_83     INTEGER NULL,     -- 4 bytes of data
sid_data_84     INTEGER NULL,     -- 4 bytes of data
sid_data_85     INTEGER NULL,     -- 4 bytes of data
sid_data_86     INTEGER NULL,     -- 4 bytes of data
sid_data_87     INTEGER NULL,     -- 4 bytes of data
sid_data_88     INTEGER NULL,     -- 4 bytes of data
sid_data_89     INTEGER NULL,     -- 4 bytes of data
sid_data_90     INTEGER NULL,     -- 4 bytes of data
```

```
sid_data_91     INTEGER NULL,    -- 4 bytes of data
sid_data_92     INTEGER NULL,    -- 4 bytes of data
sid_data_93     INTEGER NULL,    -- 4 bytes of data
sid_data_94     INTEGER NULL,    -- 4 bytes of data
sid_data_95     INTEGER NULL,    -- 4 bytes of data
sid_data_96     INTEGER NULL,    -- 4 bytes of data
sid_data_97     INTEGER NULL,    -- 4 bytes of data
sid_data_98     INTEGER NULL,    -- 4 bytes of data
sid_data_99     INTEGER NULL,    -- 4 bytes of data
sid_data_100    INTEGER NULL,    -- 4 bytes of data
sid_data_101    INTEGER NULL,    -- 4 bytes of data
sid_data_102    INTEGER NULL,    -- 4 bytes of data
sid_data_103    INTEGER NULL,    -- 4 bytes of data
sid_data_104    INTEGER NULL,    -- 4 bytes of data
sid_data_105    INTEGER NULL,    -- 4 bytes of data
sid_data_106    INTEGER NULL,    -- 4 bytes of data
sid_data_107    INTEGER NULL,    -- 4 bytes of data
sid_data_108    INTEGER NULL,    -- 4 bytes of data
sid_data_109    INTEGER NULL,    -- 4 bytes of data
sid_data_110    INTEGER NULL,    -- 4 bytes of data
sid_data_111    INTEGER NULL,    -- 4 bytes of data
sid_data_112    INTEGER NULL,    -- 4 bytes of data
sid_data_113    INTEGER NULL,    -- 4 bytes of data
sid_data_114    INTEGER NULL,    -- 4 bytes of data
sid_data_115    INTEGER NULL,    -- 4 bytes of data
sid_data_116    INTEGER NULL,    -- 4 bytes of data
sid_data_117    INTEGER NULL,    -- 4 bytes of data
sid_data_118    INTEGER NULL,    -- 4 bytes of data
sid_data_119    INTEGER NULL,    -- 4 bytes of data
sid_data_120    INTEGER NULL,    -- 4 bytes of data
sid_data_121    INTEGER NULL,    -- 4 bytes of data
sid_data_122    INTEGER NULL,    -- 4 bytes of data
sid_data_123    INTEGER NULL,    -- 4 bytes of data
sid_data_124    INTEGER NULL,    -- 4 bytes of data
sid_data_125    INTEGER NULL,    -- 4 bytes of data
sid_data_126    INTEGER NULL,    -- 4 bytes of data
sid_data_127    INTEGER NULL,    -- 4 bytes of data
sid_data_128    INTEGER NULL,    -- 4 bytes of data
sid_data_129    INTEGER NULL,    -- 4 bytes of data
sid_data_130    INTEGER NULL,    -- 4 bytes of data
sid_data_131    INTEGER NULL,    -- 4 bytes of data
sid_data_132    INTEGER NULL,    -- 4 bytes of data
sid_data_133    INTEGER NULL,    -- 4 bytes of data
sid_data_134    INTEGER NULL,    -- 4 bytes of data
sid_data_135    INTEGER NULL,    -- 4 bytes of data
sid_data_136    INTEGER NULL,    -- 4 bytes of data
sid_data_137    INTEGER NULL,    -- 4 bytes of data
sid_data_138    INTEGER NULL,    -- 4 bytes of data
sid_data_139    INTEGER NULL,    -- 4 bytes of data
sid_data_140    INTEGER NULL,    -- 4 bytes of data
sid_data_141    INTEGER NULL,    -- 4 bytes of data
sid_data_142    INTEGER NULL,    -- 4 bytes of data
sid_data_143    INTEGER NULL,    -- 4 bytes of data
sid_data_144    INTEGER NULL,    -- 4 bytes of data
sid_data_145    INTEGER NULL,    -- 4 bytes of data
sid_data_146    INTEGER NULL,    -- 4 bytes of data
sid_data_147    INTEGER NULL,    -- 4 bytes of data
sid_data_148    INTEGER NULL,    -- 4 bytes of data
sid_data_149    INTEGER NULL,    -- 4 bytes of data
sid_data_150    INTEGER NULL,    -- 4 bytes of data
sid_data_151    INTEGER NULL,    -- 4 bytes of data
sid_data_152    INTEGER NULL,    -- 4 bytes of data
sid_data_153    INTEGER NULL,    -- 4 bytes of data
sid_data_154    INTEGER NULL,    -- 4 bytes of data
sid_data_155    INTEGER NULL,    -- 4 bytes of data
```

```
  sid_data_156          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_157          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_158          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_159          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_160          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_161          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_162          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_163          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_164          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_165          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_166          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_167          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_168          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_169          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_170          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_171          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_172          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_173          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_174          INTEGER NULL                                    -- 4 bytes of data
  sid_data_175          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_176          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_177          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_178          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_179          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_180          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_181          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_182          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_183          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_184          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_185          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_186          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_187          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_188          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_189          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_190          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_191          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_192          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_193          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_194          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_195          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_196          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_197          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_198          INTEGER NULL,                                   -- 4 bytes of data
  sid_data_199          INTEGER NULL                                    -- 4 bytes of data
);
  CREATE INDEX          sid_panel_id ON siddnldpriority ( sid_panel_id );
  CREATE INDEX          sid_timestamp ON siddnldpriority ( sid_timestamp );
```

CREATE TABLE signature

```
(
  sg_cardholder_id      INTEGER,                                        -- cardholder ID
  sg_create_date        DATETIME,                                       -- date signature saved
  sg_image              IMAGE,                                          -- signature image data
  sg_instance           INTEGER,                                        -- signature instance
  site                  CHAR(32) NOT NULL,                              -- Site Name
  sg_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
  sg_checksum           INTEGER NOT NULL,                               -- Record checksum
  PRIMARY KEY           ( sg_cardholder_id, sg_instance )
);
```

CREATE TABLE signature_save

```
(
  sg_cardholder_id      INTEGER,                                                    -- cardholder ID
  sg_create_date        DATETIME,                                                   -- date signature saved
  sg_image              IMAGE,                                                      -- signature image data
  sg_instance           INTEGER,                                                    -- signature instance
  site                  CHAR(32) NOT NULL,                                          -- Site Name
  sg_guid               uniqueidentifier NOT NULL,                                  -- Global unique record identifier
  sg_checksum           INTEGER NOT NULL,                                           -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                                  -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                          -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                           -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY           ( sg_cardholder_id, sg_instance, upd_mode desc, upd_guid )
);
```

CREATE TABLE site

```
(
  s_visitor_hrs         SMALLINT,           -- default valid hours for visitors
  s_audit_time          INTEGER,            -- time to keep audit history (in mins)
  s_alarm_time          INTEGER,            -- time to keep alarm history (in mins)
  s_xaction_time        INTEGER,            -- time to keep xaction history (in mins)
  s_comms_server        CHAR(255) NOT NULL, -- name of comms server computer
  s_printing            SMALLINT,           -- 1 = real time printing enabled
  s_printer_name        CHAR(255) NULL,     -- name of real time printer
  s_print_audit         SMALLINT,           -- 1 = print audit msgs
  s_print_host          SMALLINT,           -- 1 = print host msgs
  s_print_panel         SMALLINT,           -- 1 = print panel msgs
  s_print_alarm         SMALLINT,           -- 1 = print alarm msgs
  s_print_deny          SMALLINT,           -- 1 = print deny msgs
  s_print_grant         SMALLINT,           -- 1 = print grant msgs
  s_print_trace         SMALLINT,           -- 1 = print trace msgs
  s_backup_device       CHAR(64) NULL,      -- name of backup device as configured in SQL Server
  s_rras_server         CHAR(255) NULL,     -- name of rras server computer
  s_query_string        CHAR(64) NULL,      -- query string entered by user
  s_host_event_priority SMALLINT NULL,      -- Priority to use when sending BACnet Host Event alarms and messages
  s_host_log_priority   SMALLINT NULL,      -- Priority to use when sending BACnet Host Log alarms and messages
  s_host_logic_priority SMALLINT NULL,      -- Priority to use when sending BACnet Host Logic alarms and messages
  s_audit_priority      SMALLINT NULL,      -- Priority to use when sending BACnet Audit alarms and messages
  s_hardware_priority   SMALLINT NULL,      -- Priority to use when sending BACnet Hardware alarms and messages
  s_panel_event_priority SMALLINT NULL,     -- Priority to use when sending BACnet Panel Event alarms and messages
  s_output_priority     SMALLINT NULL,      -- Priority to use when sending BACnet Output alarms and messages
  s_grant_priority      SMALLINT NULL,      -- Priority to use when sending BACnet Grant alarms and messages
  s_deny_priority       SMALLINT NULL,      -- Priority to use when sending BACnet Deny alarms and messages
  s_trace_priority      SMALLINT NULL,      -- Priority to use when sending BACnet Trace alarms and messages
  s_ta_priority         SMALLINT NULL,      -- Priority to use when sending BACnet Time and Attendance alarms and
                                            messages
  s_bacnet_ip           CHAR(16) NULL,      -- IP address used for BACnet communications
  s_bacnet_port         INTEGER NULL,       -- IP port for BACnet communications
  s_bacnet_network      INTEGER NULL,       -- BACnet network number
  s_internal_network    INTEGER NULL,       -- BACnet internal network number
  s_badge_limit         SMALLINT,           -- If 1, badge numbers are limited to 1 - 65535, the range for D620
  s_note_time           INTEGER,            -- time to keep tour notes (in mins)
  s_floor_no            SMALLINT NULL,      -- floor number for elevator (1 - 128)
  s_door_no             SMALLINT NULL,      -- door number for cabinet(1 -128)
  s_queue_for_disabled  SMALLINT NOT NULL,  -- Download data will be queued for disabled panels if 1. No queueing takes
                                            place if 0. This flag does not control queueing of door, output or
                                            relay commands, which are never sent to a disabled panel
  s_print_elevator      SMALLINT,           -- 1 = print audit msgs
  s_print_cabinet       SMALLINT,           -- 1 = print audit msgs
  s_print_guardtour     SMALLINT,           -- 1 = print audit msgs
  s_set_pw_char         SMALLINT,           -- 1 = set pw char displayed
```

```
s_muster_time                   INTEGER NOT NULL,       -- Time to keep saved muster data in minutes
s_print_area                    SMALLINT,               -- 1 = print area msgs
s_print_muster                  SMALLINT,               -- 1 = print mustering msgs
s_redundancy_serial_port        SMALLINT NOT NULL,      -- Serial port # to use for redundancy switchover of serial loops: 0 = none
s_legacy_accgrp_dnld_disable    SMALLINT,               -- 1 = do not auto download acc grps to Legacy panels
s_dnld_delay                    SMALLINT,               -- minutes to delay download
s_max_badge_number              INTEGER NOT NULL,       -- Max Badge Number allowed
s_max_badge_issue_level         SMALLINT NOT NULL,      -- Max Badge issue level
s_max_accgrps_per_badge         SMALLINT NOT NULL,      -- Max access groups per badge
s_timepairs_per_day             SMALLINT NOT NULL,      -- Timezone time pairs per day
s_timepairs_per_tz              SMALLINT NOT NULL,      -- Timezone time pairs per timezone
s_panel_types                   INTEGER NOT NULL,       -- Bitmask of allowed  panel types
s_site_codes_0                  INTEGER NOT NULL,       -- Allowed Site Codes
s_site_code_names_0             CHAR(64) NULL,          -- name of Allowed Site Codes
s_site_codes_1                  INTEGER NOT NULL,       -- Allowed Site Codes
s_site_code_names_1             CHAR(64) NULL,          -- name of Allowed Site Codes
s_site_codes_2                  INTEGER NOT NULL,       -- Allowed Site Codes
s_site_code_names_2             CHAR(64) NULL,          -- name of Allowed Site Codes
s_site_codes_3                  INTEGER NOT NULL,       -- Allowed Site Codes
s_site_code_names_3             CHAR(64) NULL,          -- name of Allowed Site Codes
s_site_codes_4                  INTEGER NOT NULL,       -- Allowed Site Codes
s_site_code_names_4             CHAR(64) NULL,          -- name of Allowed Site Codes
s_site_codes_5                  INTEGER NOT NULL,       -- Allowed Site Codes
s_site_code_names_5             CHAR(64) NULL,          -- name of Allowed Site Codes
s_site_codes_6                  INTEGER NOT NULL,       -- Allowed Site Codes
s_site_code_names_6             CHAR(64) NULL,          -- name of Allowed Site Codes
s_site_codes_7                  INTEGER NOT NULL,       -- Allowed Site Codes
s_site_code_names_7             CHAR(64) NULL,          -- name of Allowed Site Codes
s_max_security_level            SMALLINT NOT NULL,      -- Max security issue level
s_term_with_tz                  SMALLINT NOT NULL,      -- If 1, terminals in access groups can have timezone associated
s_special_access_a_desc         CHAR(32) NOT NULL,      -- Special Access A
s_special_access_b_desc         CHAR(32) NOT NULL,      -- Special Access B
s_special_access_c_desc         CHAR(32) NOT NULL,      -- Special Access C
s_mis_export_image_folder       VARCHAR(500) NULL,
s_badge_block_xaction           SMALLINT NOT NULL,      -- If 1, badge block transaction check will be displayed
s_mfg_id                        INTEGER REFERENCES msgfiltergroup(mfg_id)
                                  NOT FOR REPLICATION,  -- message filter group ID
s_alarmstate_local              SMALLINT NOT NULL,      -- alarm state change for local alarm 0=don't accept, 1=accept
s_alarmstate_remote             SMALLINT NOT NULL,      -- alarm state change for remote alarm 0=don't accept, 1=accept
s_enable_remote_message         SMALLINT NOT NULL,      -- 1 = enable remote messages
s_max_inactive_days             INTEGER NOT NULL,       -- max days a badge may be inactive
s_return_addr                   CHAR(64),               -- return address for emails
s_smtp_server                   CHAR(64),               -- smtp server for emails
s_ras_name                      CHAR(32),               -- name of ras connection for emails
s_ras_username                  CHAR(32),               -- ras connection username
s_ras_password                  CHAR(32),               -- ras connection password
s_auth_smtp                     SMALLINT NOT NULL,      -- 1 = use authenticated SMTP
s_fda_records_enabled           SMALLINT NOT NULL,      -- fda record retention / safety policy enabled
s_fda_retention_period          SMALLINT NOT NULL,      -- record retention period in years
s_fda_warning_period            SMALLINT NOT NULL,      -- retention violation period in days
s_fda_last_backup               DATETIME NOT NULL,      -- date when last backup has been taken
s_fda_pwd_enabled               SMALLINT NOT NULL,      -- FDA Rules are applied for password management
s_max_invalid_attempt           SMALLINT NOT NULL,      -- maximum number of consecutive invalid attempts for a user account
                                                          0 = disabled
s_pwd_duration                  INTEGER NOT NULL,       -- 0 = password expires after x seconds (0 = not activated)
s_pwd_min_length                SMALLINT NOT NULL,      -- minimum length for password
s_pwd_min_alpha                 SMALLINT NOT NULL,      -- minimum number of alpha char ('A' ... 'Z', 'a' ... 'z')
s_pwd_min_numeric               SMALLINT NOT NULL,      -- minimum number of numeric char ('0' ... '9')
s_pwd_min_other                 SMALLINT NOT NULL,      -- minimum number of non-alpha and non-numeric characters
s_redn_time                     INTEGER NOT NULL,       -- time to keep redundancy history (in mins)
s_exttrig_database              SMALLINT NOT NULL,      -- 1 = check database for external event triggers
s_exttrig_ip                    SMALLINT NOT NULL,      -- 1 = check ip port for external event triggers
s_exttrig_rs232                 SMALLINT NOT NULL,      -- 1 = check rs232 for external event triggers
s_exttrig_file                  SMALLINT NOT NULL,      -- 1 = check file for external event triggers
s_exttrig_rs232_portnum         SMALLINT NOT NULL,      -- RS232 Com Port Number(1: COM1, 2: COM2....)
```

```
  s_exttrig_rs232_baudrate          INTEGER NOT NULL,          -- RS232 Baudrate
  s_exttrig_rs232_parity            SMALLINT NOT NULL,         -- RS232 Parity (0: Even, 1: Odd, 2: None, 3: Mark, 4: Space)
  s_exttrig_rs232_stopbits          SMALLINT NOT NULL,         -- RS232 Stopbits (0: 1, 1: 1.5, 2: 2)
  s_exttrig_extfile_filename        CHAR(255),                 -- External File check Filename
  s_exttrig_extfile_scaninterval    INTEGER NOT NULL,          -- External File scanning interval in seconds
  s_exttrig_extdb_scaninterval      INTEGER NOT NULL,          -- External DB scanning interval in seconds
  s_global_soft_inxit               SMALLINT NOT NULL,         -- 1 = perform global Soft In-X-It tracking
  site                              CHAR(32)PRIMARY KEY,       -- Site Name
  s_guid                            uniqueidentifier ROWGUIDCOL
                                      UNIQUE NOT NULL,         -- Global unique record identifier
  s_checksum                        INTEGER NOT NULL,          -- Record checksum
  s_rms_no_msg_receive              INTEGER NOT NULL,          -- Alarm Period 'Communication error receiving' (0= no monitoring)
  s_rms_no_msg_transmit             INTEGER NOT NULL,          -- Alarm Period 'Communication error transmitting' (0 = no monitoring)
  s_print_av                        SMALLINT,                  -- 1 = print audit msgs
  s_badge_style                     SMALLINT NOT NULL,         -- 0 = Normal
  s_fascn_agency_default            SMALLINT NOT NULL,         -- default value for Agency field for FASCN badges
  s_fascn_system_default            SMALLINT NOT NULL,         -- default value for System field for FASCN badges
  s_fascn_series_default            SMALLINT NOT NULL,         -- default value for Series field for FASCN badges
  s_request_queue_time              INTEGER NOT NULL,          -- time to keep request queue history (in mins)
  s_dnld_badge_undefined            SMALLINT NOT NULL,         -- 1 = always download badges with entry/exit status of UNDEFINED
  s_shared_dnld_accgrp              SMALLINT NOT NULL,         -- 1 = download acc grp when downloading badges in shared mode
  s_global_inxit_sync               SMALLINT NOT NULL,         -- 1 = sync badge inxit to all panels
  s_request_approver_levels         SMALLINT NOT NULL,         -- number of approvers for a request (0-3)
  s_request_approver_emails         SMALLINT NOT NULL,         -- 1 = send email to request approvers
  s_approver_superuser_override     SMALLINT NOT NULL,         -- 1 = Super User rules override partition rules
  s_request_expiration_days         SMALLINT NOT NULL,         -- number of days that a request is valid
  s_hello_domain                    CHAR(64),                  -- domain name used in SMTP hello command
  s_print_intrusion                 SMALLINT,                  -- 1 = print intrusion msgs
  s_xmlrpc_password_mode            INT NOT NULL,              -- password decryption mode for XmlRpc methods
  s_xmlrpc_rsa_key                  CHAR(32),                  -- RSA decryption key for XmlRpc
  s_xmlrpc_any_ip                   SMALLINT NOT NULL,         -- 1 allow any IP address
  s_traced_alarm_granted_access     SMALLINT NOT NULL,         -- 1 allow alarm generated for granted access
  s_traced_alarm_denied_access      SMALLINT NOT NULL,         -- 1 allow alarm generated for denied access
  s_ads_path                        VARCHAR(500) NULL,         -- Active Directory path
  s_delayed_downloads               SMALLINT NOT NULL,         -- 1 enable delayed downloads
  s_ads_name_formatting             VARCHAR(256) NULL,         -- Active Directory username formatting
  s_ads_options                     INT NOT NULL,              -- Active Directory option flags
  s_webaccess_ads_enable            SMALLINT NOT NULL,         -- 1 enable Active Directory login for WebAccess
  s_webaccess_ads_partition         INTEGER REFERENCES partition(part_number)
                                      NOT FOR REPLICATION,     -- partition to use for Active Directory login for WebAccess
  s_webaccess_ads_uistyle           VARCHAR(64) NULL,          -- UI Style to use for Active Directory login for WebAccess
  s_webaccess_ads_privgrp           INTEGER REFERENCES privgrp(pg_id)
                                      NOT FOR REPLICATION,     -- privgrp to use for Active Directory login for WebAccess
  s_print_firealarm                 SMALLINT,                  -- 1 = print
  s_crosssite_accgrp_edit           SMALLINT NOT NULL,         -- 1 = allow across site access group editing
  s_max_pin_code_digits             SMALLINT NOT NULL,         -- 4 to 9 digits
  s_max_visitor_hrs                 SMALLINT NOT NULL,         -- max valid hours for visitors
  s_term_unknown_when_panel_down    SMALLINT NOT NULL,         -- 1 = set terminals unknown when panel down
  s_bacnet_enabled                  SMALLINT NOT NULL,         -- 1 = BACnet interface enabled
);
```

CREATE TABLE sitepartmap

```
(
  pm_am_guid          uniqueidentifier REFERENCES p2kadxmap(am_Guid)
                        NOT FOR REPLICATION NOT NULL,          -- Site director's ip address
  pm_Partition        INTEGER REFERENCES partition(part_number)
                        NOT FOR REPLICATION NOT NULL,          -- ID of partition
  site                CHAR (32) NOT NULL,                    -- Site Name
  pm_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  pm_checksum         INT NULL,                              -- Record checksum
  PRIMARY KEY         ( pm_am_guid, pm_Partition, site )
);
```

CREATE TABLE sitepartmap_save

```
(
  pm_am_guid          uniqueidentifier NOT NULL,                                  -- Site director's ip address
  pm_Partition        INTEGER NOT NULL,                                           -- ID of partition
  site                CHAR (32) NOT NULL,                                         -- Site Name
  pm_guid             uniqueidentifier NOT NULL,                                  -- Global unique record identifier
  pm_checksum         INT NOT NULL,                                               -- Record checksum
  upd_guid            uniqueidentifier NOT NULL,                                  -- Guid for update operation
  upd_mode            SMALLINT NOT NULL,                                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp       DATETIME NOT NULL,                                          -- Date / Time of the update
  upd_checksum        INTEGER NOT NULL,                                           -- Checksum
  upd_rowguid         uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY         ( pm_am_guid, pm_Partition, site, upd_mode desc, upd_guid )
);
```

CREATE TABLE site_save

```
(
  s_visitor_hrs           SMALLINT,           -- default valid hours for visitors
  s_audit_time            INTEGER,            -- time to keep audit history (in mins)
  s_alarm_time            INTEGER,            -- time to keep alarm history (in mins)
  s_xaction_time          INTEGER,            -- time to keep xaction history (in mins)
  s_comms_server          CHAR(255) NOT NULL, -- name of comms server computer
  s_printing              SMALLINT,           -- 1 = real time printing enabled
  s_printer_name          CHAR(255) NULL,     -- name of real time printer
  s_print_audit           SMALLINT,           -- 1 = print audit msgs
  s_print_host            SMALLINT,           -- 1 = print host msgs
  s_print_panel           SMALLINT,           -- 1 = print panel msgs
  s_print_alarm           SMALLINT,           -- 1 = print alarm msgs
  s_print_deny            SMALLINT,           -- 1 = print deny msgs
  s_print_grant           SMALLINT,           -- 1 = print grant msgs
  s_print_trace           SMALLINT,           -- 1 = print trace msgs
  s_backup_device         CHAR(64) NULL,      -- name of backup device as configured in SQL Server
  s_rras_server           CHAR(255) NULL,     -- name of rras server computer
  s_query_string          CHAR(64) NULL,      -- query string entered by user
  s_host_event_priority   SMALLINT NULL,      -- Priority to use when sending BACnet Host Event alarms and messages
  s_host_log_priority     SMALLINT NULL,      -- Priority to use when sending BACnet Host Log alarms and messages
  s_host_logic_priority   SMALLINT NULL,      -- Priority to use when sending BACnet Host Logic alarms and messages
  s_audit_priority        SMALLINT NULL,      -- Priority to use when sending BACnet Audit alarms and messages
  s_hardware_priority     SMALLINT NULL,      -- Priority to use when sending BACnet Hardware alarms and messages
  s_panel_event_priority  SMALLINT NULL,      -- Priority to use when sending BACnet Panel Event alarms and messages
  s_output_priority       SMALLINT NULL,      -- Priority to use when sending BACnet Output alarms and messages
  s_grant_priority        SMALLINT NULL,      -- Priority to use when sending BACnet Grant alarms and messages
  s_deny_priority         SMALLINT NULL,      -- Priority to use when sending BACnet Deny alarms and messages
  s_trace_priority        SMALLINT NULL,      -- Priority to use when sending BACnet Trace alarms and messages
  s_ta_priority           SMALLINT NULL,      -- Priority to use when sending BACnet Time and Attendance alarms and
                                                 messages
  s_bacnet_ip             CHAR(16) NULL,      -- IP address used for BACnet communications
  s_bacnet_port           INTEGER NULL,       -- IP port for BACnet communications
  s_bacnet_network        INTEGER NULL,       -- BACnet network number
  s_internal_network      INTEGER NULL,       -- BACnet internal network number
  s_badge_limit           SMALLINT,           -- If 1, badge numbers are limited to 1 - 65535, the range for D620
  s_note_time             INTEGER,            -- time to keep tour notes (in mins)
  s_floor_no              SMALLINT NULL,      -- floor number for elevator (1 - 128)
  s_door_no               SMALLINT NULL,      -- door number for cabinet(1 -128)
  s_queue_for_disabled    SMALLINT NOT NULL,  -- Download data will be queued for disabled panels if 1. No queueing
                                                 takes place if 0. This flag does not control queueing of door, output or
                                                 relay commands, which are never sent to a disabled panel
  s_print_elevator        SMALLINT,           -- 1 = print audit msgs
  s_print_cabinet         SMALLINT,           -- 1 = print audit msgs
  s_print_guardtour       SMALLINT,           -- 1 = print audit msgs
  s_set_pw_char           SMALLINT,           -- 1 = set pw char displayed
  s_muster_time           INTEGER NOT NULL,   -- Time to keep saved muster data in minutes
  s_print_area            SMALLINT,           -- 1 = print area msgs
```

```
s_print_muster                  SMALLINT,                   -- 1 = print mustering msgs
s_redundancy_serial_port        SMALLINT NOT NULL,          -- Serial port # to use for redundancy switchover of serial loops: 0 = none
s_legacy_accgrp_dnld_disable    SMALLINT,                   -- 1 = do not auto download acc grps to Legacy panels
s_dnld_delay                    SMALLINT,                   -- minutes to delay download
s_max_badge_number              INTEGER  NOT NULL,          -- Max Badge Number allowed
s_max_badge_issue_level         SMALLINT NOT NULL,          -- Max Badge issue level
s_max_accgrps_per_badge         SMALLINT NOT NULL,          -- Max access groups per badge
s_timepairs_per_day             SMALLINT NOT NULL,          -- Timezone time pairs per day
s_timepairs_per_tz              SMALLINT NOT NULL,          -- Timezone time pairs per timezone
s_panel_types                   INTEGER  NOT NULL,          -- Bitmask of allowed  panel types
s_site_codes_0                  INTEGER NOT NULL,           -- Allowed Site Codes
s_site_code_names_0             CHAR(64) NULL,              -- name of Allowed Site Codes
s_site_codes_1                  INTEGER NOT NULL,           -- Allowed Site Codes
s_site_code_names_1             CHAR(64) NULL,              -- name of Allowed Site Codes
s_site_codes_2                  INTEGER NOT NULL,           -- Allowed Site Codes
s_site_code_names_2             CHAR(64) NULL,              -- name of Allowed Site Codes
s_site_codes_3                  INTEGER NOT NULL,           -- Allowed Site Codes
s_site_code_names_3             CHAR(64) NULL,              -- name of Allowed Site Codes
s_site_codes_4                  INTEGER NOT NULL,           -- Allowed Site Codes
s_site_code_names_4             CHAR(64) NULL,              -- name of Allowed Site Codes
s_site_codes_5                  INTEGER NOT NULL,           -- Allowed Site Codes
s_site_code_names_5             CHAR(64) NULL,              -- name of Allowed Site Codes
s_site_codes_6                  INTEGER NOT NULL,           -- Allowed Site Codes
s_site_code_names_6             CHAR(64) NULL,              -- name of Allowed Site Codes
s_site_codes_7                  INTEGER NOT NULL,           -- Allowed Site Codes
s_site_code_names_7             CHAR(64) NULL,              -- name of Allowed Site Codes
s_max_security_level            SMALLINT NOT NULL,          -- Max security issue level
s_term_with_tz                  SMALLINT NOT NULL,          -- If 1, terminals in access groups can have timezone associated
s_special_access_a_desc         CHAR(32) NOT NULL,          -- Special Access A
s_special_access_b_desc         CHAR(32) NOT NULL,          -- Special Access B
s_special_access_c_desc         CHAR(32) NOT NULL,          -- Special Access C
s_mis_export_image_folder       VARCHAR(500) NULL,
s_badge_block_xaction           SMALLINT NOT NULL,          -- If 1, badge block transaction check will be displayed
s_mfg_id                        INTEGER,                    -- message filter group ID
s_alarmstate_local              SMALLINT NOT NULL,          -- alarm state change for local alarm 0=don't accept, 1=accept
s_alarmstate_remote             SMALLINT NOT NULL,          -- alarm state change for remote alarm 0=don't accept, 1=accept
s_enable_remote_message         SMALLINT NOT NULL,          -- 1 = enable remote messages
s_max_inactive_days             INTEGER NOT NULL,           -- max days a badge may be inactive
s_return_addr                   CHAR(64),                   -- return address for emails
s_smtp_server                   CHAR(64),                   -- smtp server for emails
s_ras_name                      CHAR(32),                   -- name of ras connection for emails
s_ras_username                  CHAR(32),                   -- ras connection username
s_ras_password                  CHAR(32),                   -- ras connection password
s_auth_smtp                     SMALLINT NOT NULL,          -- 1 = use authenticated SMTP
s_fda_records_enabled           SMALLINT NOT NULL,          -- fda record retention / safety policy enabled
s_fda_retention_period          INTEGER  NOT NULL,          -- record retention period in years
s_fda_warning_period            INTEGER  NOT NULL,          -- retention violation period in days
s_fda_last_backup               DATETIME NOT NULL,          -- date when last backup has been taken
s_fda_pwd_enabled               SMALLINT NOT NULL,          -- FDA Rules are applied for password management
s_max_invalid_attempt           SMALLINT NOT NULL,          -- maximum number of consecutive invalid attempts for a user account
                                                              0 = disabled

s_pwd_duration                  INTEGER NOT NULL,           -- 0 = password expires after x seconds (0 = not activated)
s_pwd_min_length                SMALLINT NOT NULL,          -- minimum length for password
s_pwd_min_alpha                 SMALLINT NOT NULL,          -- minimum number of alpha char ('A' ... 'Z', 'a' ... 'z')
s_pwd_min_numeric               SMALLINT NOT NULL,          -- minimum number of numeric char ('0' ... '9')
s_pwd_min_other                 SMALLINT NOT NULL,          -- minimum number of non-alpha and non-numeric characters
s_redn_time                     INTEGER,                    -- time to keep redundancy history (in mins)
s_exttrig_database              SMALLINT NOT NULL,          -- 1 = check database for external event triggers
s_exttrig_ip                    SMALLINT NOT NULL,          -- 1 = check ip port for external event triggers
s_exttrig_rs232                 SMALLINT NOT NULL,          -- 1 = check rs232 for external event triggers
s_exttrig_file                  SMALLINT NOT NULL,          -- 1 = check file for external event triggers
s_exttrig_rs232_portnum         SMALLINT NOT NULL,          -- RS232 Com Port Number(1: COM1, 2: COM2....)
s_exttrig_rs232_baudrate        INTEGER NOT NULL,           -- RS232 Baudrate
s_exttrig_rs232_parity          SMALLINT NOT NULL,          -- RS232 Parity  (0: Even, 1: Odd, 2: None, 3: Mark, 4: Space)
s_exttrig_rs232_stopbits        SMALLINT NOT NULL,          -- RS232 Stopbits (0: 1, 1: 1.5, 2: 2)
```

```
  s_exttrig_extfile_filename        CHAR(255),                    -- External File check Filename
  s_exttrig_extfile_scaninterval    INTEGER NOT NULL,             -- External File scanning interval in seconds
  s_exttrig_extdb_scaninterval      INTEGER NOT NULL,             -- External DB scanning interval in seconds
  s_global_soft_inxit               SMALLINT NOT NULL,            -- 1 = perform global Soft In-X-It tracking
  site                              CHAR(32) NOT NULL,            -- Site Name
  s_guid                            uniqueidentifier NOT NULL,    -- Global unique record identifier
  s_checksum                        INTEGER NOT NULL,             -- Record checksum
  upd_guid                          uniqueidentifier NOT NULL,    -- Guid for update operation
  upd_mode                          SMALLINT NOT NULL,            -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                     DATETIME NOT NULL,            -- Date / Time of the update
  upd_checksum                      INTEGER NOT NULL,             -- Checksum
  s_rms_no_msg_receive              INTEGER NOT NULL,             -- Alarm Period 'Communication error receiving' (0= no monitoring)
  s_rms_no_msg_transmit             INTEGER NOT NULL,             -- Alarm Period 'Communication error transmitting' (0 = no monitoring)
  upd_rowguid                       uniqueidentifier ROWGUIDCOL DEFAULT newid()
                                       UNIQUE NOT NULL,           -- Global unique record identifier
  s_print_av                        SMALLINT,                     -- 1 = print audit msgs
  s_badge_style                     SMALLINT,                     -- 0 = Normal
  s_fascn_agency_default            SMALLINT,                     -- default value for Agency field for FASCN badges
  s_fascn_system_default            SMALLINT,                     -- default value for System field for FASCN badges
  s_fascn_series_default            SMALLINT,                     -- default value for Series field for FASCN badges
  s_request_queue_time              INTEGER,                      -- time to keep request queue history (in mins)
  s_dnld_badge_undefined            SMALLINT,                     -- 1 = always download badges with entry/exit status of UNDEFINED
  s_shared_dnld_accgrp              SMALLINT,                     -- 1 = download acc grp when downloading badges in shared mode
  s_global_inxit_sync               SMALLINT,                     -- 1 = sync badge inxit to all panels
  s_request_approver_levels         SMALLINT,                     -- number of approvers for a request (0-3)
  s_request_approver_emails         SMALLINT,                     -- 1 = send email to request approvers
  s_approver_superuser_override     SMALLINT,                     -- 1 = Super User rules override partition rules
  s_request_expiration_days         SMALLINT,                     -- number of days that a request is valid
  s_hello_domain                    CHAR(64),                     -- domain name used in SMTP hello command
  s_print_intrusion                 SMALLINT,                     -- 1 = print intrusion msgs
  s_xmlrpc_password_mode            INT,                          -- password decryption mode for XmlRpc methods
  s_xmlrpc_rsa_key                  CHAR(32),                     -- RSA decryption key for XmlRpc
  s_xmlrpc_any_ip                   SMALLINT,                     -- 1 allow any IP address
  s_traced_alarm_granted_access     SMALLINT NOT NULL,            -- 1 allow alarm generated for granted access
  s_traced_alarm_denied_access      SMALLINT NOT NULL,            -- 1 allow alarm generated for denied access
  s_ads_path                        VARCHAR(500) NULL,            -- Active Directory path
  s_delayed_downloads               SMALLINT,                     -- 1 enable delayed downloads
  s_ads_name_formatting             VARCHAR(256) NULL,            -- Active Directory username formatting
  s_ads_options                     INT NOT NULL,                 -- Active Directory option flags
  s_webaccess_ads_enable            SMALLINT NOT NULL,            -- 1 enable Active Directory login for WebAccess
  s_webaccess_ads_partition         INTEGER,                      -- partition to use for Active Directory login for WebAccess
  s_webaccess_ads_uistyle           VARCHAR(64) NULL,             -- UI Style to use for Active Directory login for WebAccess
  s_webaccess_ads_privgrp           INTEGER,                      -- privgrp to use for Active Directory login for WebAccess
  s_print_firealarm                 SMALLINT,                     -- 1 = print
  s_crosssite_accgrp_edit           SMALLINT NULL,                -- 1 = allow across site access group editing
  s_max_pin_code_digits             SMALLINT NULL,                -- 4 to 9 digits
  s_max_visitor_hrs                 SMALLINT NULL,                -- max valid hours for visitors
  s_term_unknown_when_panel_down    SMALLINT NULL,                -- 1 = set terminals unknown when panel down
  s_bacnet_enabled                  SMALLINT NOT NULL,            -- 1 = BACnet interface enabled
  PRIMARY KEY                       (site, upd_mode desc, upd_guid)
);
```

CREATE TABLE softalarm

```
(
  sa_panel_id                INTEGER REFERENCES panel(p_panel_id)
                                NOT FOR REPLICATION,            -- panel ID
  sa_term_index              SMALLINT,                         -- terminal index on which to report panel soft alarms
  sa_pan_batt_low            SMALLINT,                         -- 1 = panel battery low alarm enabled
  sa_pan_ac_fail             SMALLINT,                         -- 1 = panel ac fail alarm enabled
  sa_pan_tamper              SMALLINT,                         -- 1 = panel tamper alarm enabled
  sa_pan_batt_low_pt         SMALLINT,                         -- input point number for panel battery low alarm
  sa_pan_ac_fail_pt          SMALLINT,                         -- input point number for panel ac fail alarm
  sa_pan_tamper_pt           SMALLINT,                         -- input point number for panel tamper alarm
```

```
    sa_pan_batt_low_rel         SMALLINT,                       -- 1 = panel battery low alarm relay enabled (legacy panels only)
    sa_pan_ac_fail_rel          SMALLINT,                       -- 1 = panel ac fail alarm relay enabled (legacy panels only)
    sa_pan_tamper_rel           SMALLINT,                       -- 1 = panel tamper alarm relay enabled (legacy panels only)
    sa_rd_batt_low              SMALLINT,                       -- 1 = terminal battery low alarm enabled
    sa_rd_ac_fail               SMALLINT,                       -- 1 = terminal ac fail alarm enabled
    sa_rd_tamper                SMALLINT,                       -- 1 = terminal tamper alarm
    sa_rd_card_parity           SMALLINT,                       -- 1 = terminal card parity alarm
    sa_rd_pin_error             SMALLINT,                       -- 1 = terminal pin code error alarm
    sa_rd_forced_door           SMALLINT,                       -- 1 = terminal forced door alarm
    sa_rd_durress               SMALLINT,                       -- 1 = terminal duress alarm
    sa_rd_soft_inxit            SMALLINT,                       -- 1 = terminal soft_inxit alarm
    sa_rd_batt_low_pt           SMALLINT,                       -- input point number for terminal battery low alarm
    sa_rd_ac_fail_pt            SMALLINT,                       -- input point number for terminal ac fail alarm
    sa_rd_tamper_pt             SMALLINT,                       -- input point number for terminal tamper alarm
    sa_rd_card_parity_pt        SMALLINT,                       -- input point number for terminal card parity alarm
    sa_rd_pin_error_pt          SMALLINT,                       -- input point number for terminal pin code error alarm
    sa_rd_forced_door_pt        SMALLINT,                       -- input point number for terminal forced door alarm
    sa_rd_duress_pt             SMALLINT,                       -- input point number for terminal duress alarm
    sa_rd_soft_inxit_pt         SMALLINT,                       -- input point number for terminal soft_inxit alarm
    sa_rd_prop_door_pt          SMALLINT,                       -- input point number for terminal propped door alarm
    sa_rd_batt_low_rel          SMALLINT,                       -- 1 = terminal battery low alarm relay enabled
    sa_rd_ac_fail_rel           SMALLINT,                       -- 1 = terminal ac fail alarm relay enabled
    sa_rd_tamper_rel            SMALLINT,                       -- 1 = terminal tamper relay alarm
    sa_rd_card_parity_rel       SMALLINT,                       -- 1 = terminal card parity relay alarm
    sa_rd_pin_error_rel         SMALLINT,                       -- 1 = terminal pin code error relay alarm
    sa_rd_forced_door_rel       SMALLINT,                       -- 1 = terminal forced door relay alarm
    sa_rd_durress_rel           SMALLINT,                       -- 1 = terminal duress relay alarm
    sa_rd_cbatt_rel             SMALLINT,                       -- 1 = terminal card battery relay alarm
    site                        CHAR(32) NOT NULL,              -- Site Name
    sa_guid                     uniqueidentifier ROWGUIDCOL UNIQUE
                                    NOT NULL,                   -- Global unique record identifier
    sa_checksum                 INTEGER NOT NULL,               -- Record checksum
);
    CREATE UNIQUE INDEX         sa_panel_id ON softalarm ( sa_panel_id );
```

CREATE TABLE softalarm_save

```
(
    sa_panel_id                 INTEGER,                        -- panel ID
    sa_term_index               SMALLINT,                       -- terminal index on which to report panel soft alarms
    sa_pan_batt_low             SMALLINT,                       -- 1 = panel battery low alarm enabled
    sa_pan_ac_fail              SMALLINT,                       -- 1 = panel ac fail alarm enabled
    sa_pan_tamper               SMALLINT,                       -- 1 = panel tamper alarm enabled
    sa_pan_batt_low_pt          SMALLINT,                       -- input point number for panel battery low alarm
    sa_pan_ac_fail_pt           SMALLINT,                       -- input point number for panel ac fail alarm
    sa_pan_tamper_pt            SMALLINT,                       -- input point number for panel tamper alarm
    sa_pan_batt_low_rel         SMALLINT,                       -- 1 = panel battery low alarm relay enabled (legacy panels only)
    sa_pan_ac_fail_rel          SMALLINT,                       -- 1 = panel ac fail alarm relay enabled (legacy panels only)
    sa_pan_tamper_rel           SMALLINT,                       -- 1 = panel tamper alarm relay enabled (legacy panels only)
    sa_rd_batt_low              SMALLINT,                       -- 1 = terminal battery low alarm enabled
    sa_rd_ac_fail               SMALLINT,                       -- 1 = terminal ac fail alarm enabled
    sa_rd_tamper                SMALLINT,                       -- 1 = terminal tamper alarm
    sa_rd_card_parity           SMALLINT,                       -- 1 = terminal card parity alarm
    sa_rd_pin_error             SMALLINT,                       -- 1 = terminal pin code error alarm
    sa_rd_forced_door           SMALLINT,                       -- 1 = terminal forced door alarm
    sa_rd_durress               SMALLINT,                       -- 1 = terminal duress alarm
    sa_rd_soft_inxit            SMALLINT,                       -- 1 = terminal soft_inxit alarm
    sa_rd_batt_low_pt           SMALLINT,                       -- input point number for terminal battery low alarm
    sa_rd_ac_fail_pt            SMALLINT,                       -- input point number for terminal ac fail alarm
    sa_rd_tamper_pt             SMALLINT,                       -- input point number for terminal tamper alarm
    sa_rd_card_parity_pt        SMALLINT,                       -- input point number for terminal card parity alarm
    sa_rd_pin_error_pt          SMALLINT,                       -- input point number for terminal pin code error alarm
    sa_rd_forced_door_pt        SMALLINT,                       -- input point number for terminal forced door alarm
    sa_rd_duress_pt             SMALLINT,                       -- input point number for terminal duress alarm
```

```
  sa_rd_soft_inxit_pt       SMALLINT,                                   -- input point number for terminal soft_inxit alarm
  sa_rd_prop_door_pt        SMALLINT,                                   -- input point number for terminal propped door alarm
  sa_rd_batt_low_rel        SMALLINT,                                   -- 1 = terminal battery low alarm relay enabled
  sa_rd_ac_fail_rel         SMALLINT,                                   -- 1 = terminal ac fail alarm relay enabled
  sa_rd_tamper_rel          SMALLINT,                                   -- 1 = terminal tamper relay alarm
  sa_rd_card_parity_rel     SMALLINT,                                   -- 1 = terminal card parity relay alarm
  sa_rd_pin_error_rel       SMALLINT,                                   -- 1 = terminal pin code error relay alarm
  sa_rd_forced_door_rel     SMALLINT,                                   -- 1 = terminal forced door relay alarm
  sa_rd_durress_rel         SMALLINT,                                   -- 1 = terminal duress relay alarm
  sa_rd_cbatt_rel           SMALLINT,                                   -- 1 = terminal card battery relay alarm
  site                      CHAR(32) NOT NULL,                          -- Site Name
  sa_guid                   uniqueidentifier NOT NULL,                  -- Global unique record identifier
  sa_checksum               INTEGER NOT NULL,                           -- Record checksum
  upd_guid                  uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode                  SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp             DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum              INTEGER NOT NULL,                           -- Checksum
  upd_rowguid               uniqueidentifier ROWGUIDCOL DEFAULT
                              newid() UNIQUE NOT NULL,                  -- Global unique record identifier
  PRIMARY KEY               (sa_panel_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE softwareupdate

```
(
  su_name                   CHAR(64) NOT NULL,                          -- name of software update
  su_major_version          INT NOT NULL,                               -- update major version
  su_minor_version          INT NOT NULL,                               -- update minor version
  su_build_version          INT NOT NULL,                               -- update build version
  su_sp_version             INT NOT NULL,                               -- service pack number (0 = release version)
  su_timestamp              DATETIME NOT NULL,                          -- time update installed on server
  su_server_path            VARCHAR(512) NOT NULL,                      -- path on server to update
  su_shared_path            VARCHAR(512) NOT NULL,                      -- shared network path to update
  su_command                CHAR(54) NOT NULL,                          -- command to execute to launch update
  su_status                 INT NOT NULL,                               -- 0 = Not available, 1 = available, 2 = required
  site                      CHAR(32) NOT NULL,                          -- Site Name
  su_guid                   uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,-- Global unique record identifier
  su_checksum               INTEGER NOT NULL,                           -- Record checksum
  PRIMARY KEY               ( site, su_major_version, su_minor_version, su_build_version, su_sp_version, su_timestamp )
);
  CREATE INDEX              su_status ON softwareupdate ( su_status );
```

CREATE TABLE softwareupdate_save

```
(
  su_name                   CHAR(64) NOT NULL,                          -- name of software update
  su_major_version          INT NOT NULL,                               -- update major version
  su_minor_version          INT NOT NULL,                               -- update minor version
  su_build_version          INT NOT NULL,                               -- update build version
  su_sp_version             INT NOT NULL,                               -- service pack number (0 = release version)
  su_timestamp              DATETIME NOT NULL,                          -- time update installed on server
  su_server_path            VARCHAR(512) NOT NULL,                      -- path on server to update
  su_shared_path            VARCHAR(512) NOT NULL,                      -- shared network path to update
  su_command                CHAR(54) NOT NULL,                          -- command to execute to launch update
  su_status                 INT NOT NULL,                               -- 0 = Not available, 1 = available, 2 = required
  site                      CHAR(32) NOT NULL,                          -- Site Name
  su_guid                   uniqueidentifier NOT NULL,                  -- Global unique record identifier
  su_checksum               INTEGER NOT NULL,                           -- Record checksum
  upd_guid                  uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode                  SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp             DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum              INTEGER NOT NULL,                           -- Checksum
  upd_rowguid               uniqueidentifier ROWGUIDCOL DEFAULT newid()
                              UNIQUE NOT NULL,                          -- Global unique record identifier
```

```
  PRIMARY KEY          ( site, su_major_version, su_minor_version, su_build_version, su_sp_version, su_timestamp, upd_mode desc, upd_guid )
);
```

CREATE TABLE station

```
(
  stn_partition              INTEGER REFERENCES partition(part_number)
                               NOT FOR REPLICATION,                    -- partition ID to which this station belongs
  stn_public                 SMALLINT,                                 -- 1 = public, 0 = private
  stn_id                     INTEGER IDENTITY NOT FOR REPLICATION UNIQUE, -- unique ID for this station
  stn_name                   CHAR(32) NOT NULL,                        -- station name
  stn_tz                     INTEGER NULL REFERENCES timezone(tz_id)
                               NOT FOR REPLICATION,                    -- timezone ID when this station active (always
                                                                          active if NULL)
  stn_enable                 INTEGER,                                  -- 1 = station enabled
  stn_badge_ws               INTEGER,                                  -- 1 = station is a badging station
  stn_server                 INTEGER NOT NULL,                         -- 1 = station is also a server
  stn_mfg_id                 INTEGER REFERENCES msgfiltergroup(mfg_id)
                               NOT FOR REPLICATION,                    -- message filter group ID
  stn_alarm_ws               SMALLINT NOT NULL,                        --- Can Alarm Monitor be shutdown
  site                       CHAR(32) NOT NULL,                        -- Site Name
  stn_guid                   uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  stn_checksum               INTEGER NOT NULL,                         -- Record checksum
  stn_location               CHAR(32) NULL,                            -- station location
  stn_major_version          INT NOT NULL,                             -- current software major version
  stn_minor_version          INT NOT NULL,                             -- current software minor version
  stn_build_version          INT NOT NULL,                             -- current software build version
  stn_sp_version             INT NOT NULL,                             -- current service pack number (0 = release version)
  stn_station_timestamp      DATETIME NOT NULL,                        -- time update installed on this station
  PRIMARY KEY                (stn_name, stn_id)
);
  CREATE UNIQUE INDEX        stn_site_name ON station (site, stn_name);
```

CREATE TABLE station_stat

```
(
  sstat_station_id           INTEGER REFERENCES station(stn_id) NOT FOR
                               REPLICATION NOT NULL,                   -- station ID
  sstat_username             CHAR(16) NULL,                            -- user name who is logged in
  sstat_logged_in            SMALLINT,                                 -- 1 = user logged in
  sstat_timestamp            DATETIME,                                 -- time user logged in
  sstat_badging              SMALLINT,                                 -- 1 = station is a badging station
  sstat_station_utctime      DATETIME,                                 -- UTC time workstation last reported itself
  sstat_session_guid         uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- session guid for this connection
  site                       CHAR(32) NOT NULL,                        -- Site Name
  sstat_guid                 uniqueidentifier PRIMARY KEY,             -- Record guid for this connection
  sstat_checksum             INTEGER NOT NULL,                         -- Record checksum
);
  CREATE INDEX               station_stat_id ON station_stat ( sstat_station_id );
  CREATE INDEX               station_stat_username ON station_stat ( sstat_username );
```

CREATE TABLE station_save

```
(
  stn_partition              INTEGER,                                  -- partition ID to which this station belongs
  stn_public                 SMALLINT,                                 -- 1 = public, 0 = private
  stn_id                     INTEGER NOT NULL,                         -- unique ID for this station
  stn_name                   CHAR(32) NOT NULL,                        -- station name
  stn_tz                     INTEGER NULL,                             -- timezone ID when this station active (always active if NULL)
  stn_enable                 INTEGER,                                  -- 1 = station enabled
  stn_badge_ws               INTEGER,                                  -- 1 = station is a badging station
  stn_server                 INTEGER NOT NULL,                         -- 1 = station is also a server
  stn_mfg_id                 INTEGER,                                  -- message filter group ID
  stn_alarm_ws               SMALLINT NOT NULL,                        --- Can Alarm Monitor be shutdown
```

```
  site                  CHAR(32) NOT NULL,                          -- Site Name
  stn_guid              uniqueidentifier NOT NULL,                  -- Global unique record identifier
  stn_checksum          INTEGER NOT NULL,                           -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                           -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                          -- Global unique record identifier
  stn_location          CHAR(32) NULL,                              -- station location
  stn_major_version     INT NULL,                                   -- current software major version
  stn_minor_version     INT NULL,                                   -- current software minor version
  stn_build_version     INT NULL,                                   -- current software build version
  stn_sp_version        INT NULL,                                   -- current service pack number (0 = release version)
  stn_station_timestamp DATETIME NULL,                              -- time update installed on this station
  PRIMARY KEY           (stn_name, stn_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE stia_access_code_door

```
(
  access_code_door_id   uniqueidentifier PRIMARY KEY NOT NULL,      -- Global unique record identifier
  access_code_id        uniqueidentifier NOT NULL,                  -- Global unique record identifier
  door_id               uniqueidentifier NOT NULL,                  -- Global unique record identifier
);
  CREATE INDEX          door_id ON stia_access_code_door(door_id);
  CREATE INDEX          access_code_id ON stia_access_code_door(access_code_id);
```

CREATE TABLE stia_badge

```
(
  badge_id              uniqueidentifier PRIMARY KEY NOT NULL,      -- Global unique record identifier
  person_id             uniqueidentifier NOT NULL,                  -- Global unique record identifier
  job_title_id          uniqueidentifier NOT NULL,                  -- Global unique record identifier
  badge_status_id       uniqueidentifier NOT NULL,                  -- Global unique record identifier
  threat_level_id       uniqueidentifier NOT NULL,                  -- Global unique record identifier
);
  CREATE INDEX          person_id ON stia_badge(person_id);
  CREATE INDEX          job_title_id ON stia_badge(job_title_id);
  CREATE INDEX          badge_status_id ON stia_badge(badge_status_id);
  CREATE INDEX          threat_level_id ON stia_badge(threat_level_id);
```

CREATE TABLE stia_badge_access

```
(
  badge_access_id       uniqueidentifier PRIMARY KEY NOT NULL,      -- Global unique record identifier
  badge_id              uniqueidentifier NOT NULL,                  -- Global unique record identifier
  access_code_id        uniqueidentifier NOT NULL,                  -- Global unique record identifier
);
  CREATE INDEX          badge_id ON stia_badge_access(badge_id);
  CREATE INDEX          access_code_id ON stia_badge_access(access_code_id);
```

CREATE TABLE stia_badge_exception

```
(
  badge_exception_id    uniqueidentifier PRIMARY KEY NOT NULL,      -- Global unique record identifier
  badge_id              uniqueidentifier NOT NULL,                  -- Global unique record identifier
  door_id               uniqueidentifier NOT NULL,                  -- Global unique record identifier
);
  CREATE INDEX          badge_id ON stia_badge_exception(badge_id);
  CREATE INDEX          door_id ON stia_badge_exception(door_id);
```

```
CREATE TABLE    stia_division
(
   division_id              uniqueidentifier PRIMARY KEY NOT NULL,          -- Global unique record identifier
   company_id               uniqueidentifier NOT NULL,                      -- Global unique record identifier
);
   CREATE INDEX             company_id ON stia_division(company_id);
```

```
CREATE TABLE    stia_image_documents
(
   image_document_id        uniqueidentifier PRIMARY KEY NOT NULL,          -- Global unique record identifier
   image_type_id            uniqueidentifier NOT NULL,                      -- Global unique record identifier
   object_id                uniqueidentifier NOT NULL,                      -- Global unique record identifier
   image_format_id          uniqueidentifier NOT NULL,                      -- Global unique record identifier
);
   CREATE INDEX             image_type_id ON stia_image_documents(image_type_id);
   CREATE INDEX             object_id ON stia_image_documents(object_id);
   CREATE INDEX             image_format_id ON stia_image_documents(image_format_id);
```

```
CREATE TABLE    stia_job_title
(
   job_title_id             uniqueidentifier PRIMARY KEY NOT NULL,          -- Global unique record identifier
   division_id              uniqueidentifier NOT NULL,                      -- Global unique record identifier
);
   CREATE INDEX             division_id ON stia_job_title(division_id);
```

```
CREATE TABLE    stia_person
(
   person_id                uniqueidentifier PRIMARY KEY NOT NULL,          -- Global unique record identifier
   eye_color_id             uniqueidentifier NOT NULL,                      -- Global unique record identifier
   hair_color_id            uniqueidentifier NOT NULL,                      -- Global unique record identifier
   race_id                  uniqueidentifier NOT NULL,                      -- Global unique record identifier
   birth_country_id         uniqueidentifier NOT NULL,                      -- Global unique record identifier
   birth_state_prov_id      uniqueidentifier NOT NULL,                      -- Global unique record identifier
);
   CREATE INDEX             eye_color_id ON stia_person(eye_color_id);
   CREATE INDEX             hair_color_id ON stia_person(hair_color_id);
   CREATE INDEX             race_id ON stia_person(race_id);
   CREATE INDEX             birth_country_id ON stia_person(birth_country_id);
   CREATE INDEX             birth_state_prov_id ON stia_person(birth_state_prov_id);
```

```
CREATE TABLE    svr_startup_cnfg
(
   svr_id                   INTEGER IDENTITY NOT FOR REPLICATION PRIMARY KEY,   -- unique database ID
   svr_servicename          varchar(32) NOT NULL,
   svr_servicetitle         varchar(48) NOT NULL,
   svr_servername           varchar(32) NOT NULL,
   svr_enable_flag          SMALLINT NOT NULL,
   svr_iResourceID          SMALLINT NOT NULL,
   svr_bInstalled           SMALLINT NOT NULL,
   svr_failurerecovery      SMALLINT NOT NULL,                              -- ID of failure recovery options
   site                     CHAR(32) NOT NULL,                              -- Site Name
   svr_guid                 uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
   svr_checksum             INTEGER NOT NULL,                               -- Record checksum
);
   CREATE UNIQUE INDEX      svr_site_name_server ON svr_startup_cnfg ( site, svr_servicename, svr_servername );
   CREATE UNIQUE INDEX      svr_site_title_server ON svr_startup_cnfg ( site, svr_servicetitle, svr_servername );
```

CREATE TABLE svr_startup_cnfg_save

```
(
  svr_id                INTEGER NOT NULL,                                          -- unique database ID
  svr_servicename       varchar(32) NOT NULL,
  svr_servicetitle      varchar(48) NOT NULL,
  svr_servername        varchar(32) NOT NULL,
  svr_enable_flag       SMALLINT NOT NULL,
  svr_iResourceID       SMALLINT NOT NULL,
  svr_bInstalled        SMALLINT NOT NULL,
  svr_failurerecovery   SMALLINT NOT NULL,                                         -- ID of failure recovery options
  site                  CHAR(32) NOT NULL,                                         -- Site Name
  svr_guid              uniqueidentifier NOT NULL,                                 -- Global unique record identifier
  svr_checksum          INTEGER NOT NULL,                                          -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                                 -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                         -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                          -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY           (svr_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE termgrp

```
(
  tg_partition          INTEGER REFERENCES partition(part_number)
                          NOT FOR REPLICATION,                                   -- partition ID to which this group belongs
  tg_public             SMALLINT,                                                -- 1 = public, 0 = private
  tg_id                 INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,             -- unique ID for this terminal group
  tg_name               CHAR(40) NOT NULL,                                       -- terminal group name
  site                  CHAR(32) NOT NULL,                                       -- Site Name
  tg_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,             -- Global unique record identifier
  tg_checksum           INTEGER NOT NULL,                                        -- Record checksum
  PRIMARY KEY           ( tg_partition, tg_id )
);
  CREATE INDEX          tg_public ON termgrp ( tg_public );
  CREATE UNIQUE INDEX   tg_site_name ON termgrp (site, tg_name);
```

CREATE TABLE termgrpterm

```
(
  tgt_tg_id             INTEGER REFERENCES termgrp(tg_id) NOT FOR REPLICATION,   -- term grp ID
  tgt_term_id           INTEGER REFERENCES terminal(tp_term_id) NOT FOR REPLICATION, -- terminal ID
  site                  CHAR(32) NOT NULL,                                       -- Site Name
  tgt_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,             -- Global unique record identifier
  tgt_checksum          INTEGER NOT NULL,                                        -- Record checksum
  PRIMARY KEY           ( tgt_tg_id, tgt_term_id)
);
  CREATE INDEX          tgt_term_id ON termgrpterm ( tgt_term_id );
```

CREATE TABLE termgrpterm_save

```
(
  tgt_tg_id             INTEGER,                                                 -- term grp ID
  tgt_term_id           INTEGER,                                                 -- terminal ID
  site                  CHAR(32) NOT NULL,                                       -- Site Name
  tgt_guid              uniqueidentifier NOT NULL,                               -- Global unique record identifier
  tgt_checksum          INTEGER NOT NULL,                                        -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                               -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                       -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                       -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                        -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY           ( tgt_tg_id, tgt_term_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE termgrp_save

```
(
  tg_partition          INTEGER,                                                    -- partition ID to which this group belongs
  tg_public             SMALLINT,                                                   -- 1 = public, 0 = private
  tg_id                 INTEGER NOT NULL,                                           -- unique ID for this terminal group
  tg_name               CHAR(40) NOT NULL,                                          -- terminal group name
  site                  CHAR(32) NOT NULL,                                          -- Site Name
  tg_guid               uniqueidentifier NOT NULL,                                  -- Global unique record identifier
  tg_checksum           INTEGER NOT NULL,                                           -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                                  -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                          -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                           -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY           ( tg_partition, tg_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE terminal

```
(
  tp_term_id            INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,  -- unique terminal ID
  tp_term_name          varchar(32) NOT NULL,                         -- name of terminal
  tp_model              SMALLINT NOT NULL,                            -- terminal Model
  tp_term_type          SMALLINT NOT NULL,                            -- terminal type
  tp_term_index         SMALLINT NOT NULL,                            -- terminal index (0-15)
  tp_partition          INTEGER REFERENCES partition(part_number)
                          NOT FOR REPLICATION,                        -- ID of partition for this terminal
  tp_public             SMALLINT NOT NULL,                            -- publication
  tp_panel_id           INTEGER NOT NULL REFERENCES panel(p_panel_id)
                          NOT FOR REPLICATION,                        -- panel id
  tp_query_string       CHAR(64),                                     -- query string enter by user
  tp_reader_flag        SMALLINT NOT NULL,                            -- 1 = terminal is a reader terminal
  tp_input_flag         SMALLINT NOT NULL,                            -- 1 = terminal is an input terminal
  tp_output_flag        SMALLINT NOT NULL,                            -- 1 = terminal is an output terminal
  tp_enb_dis            SMALLINT NOT NULL,                            -- 1 = terminal is enabled
  tp_term_tz            INTEGER NULL REFERENCES timezone(tz_id)
                          NOT FOR REPLICATION,                        -- timezone ID when enabled (always if NULL)
  tp_override_tz        INTEGER NULL REFERENCES timezone(tz_id)
                          NOT FOR REPLICATION,                        -- timezone ID when override (always if NULL)
  tp_pin_enable_flag    SMALLINT NOT NULL,                            -- pin pad enable/disable
  tp_pin_enable_tz      INTEGER NULL REFERENCES timezone(tz_id)
                          NOT FOR REPLICATION,                        -- timezone ID when pin code suppressed (always if NULL)
  tp_pin_disable_tz     INTEGER NULL REFERENCES timezone(tz_id)
                          NOT FOR REPLICATION,                        -- pin pad disable time zone
  tp_access_time        SMALLINT NOT NULL,                            -- access time in secs (0-25)
  tp_shunt_time         SMALLINT NOT NULL,                            -- shunt time in secs (0-255)
  tp_manual_rdr         SMALLINT NOT NULL,                            -- 1 = valid and unauthorized enabled
  tp_anti_passback      SMALLINT NOT NULL,                            -- 1 = anti-passback enabled
  tp_anti_pb_time       SMALLINT NOT NULL,                            -- anti-passback time in mins (0-60)
  tp_ntr_xit_time       SMALLINT NOT NULL,                            -- entry/exit time in secs (0-255)
  tp_soft_inxit         SMALLINT NOT NULL,                            -- 1 = soft entry/exit enabled
  tp_access_modes       SMALLINT NOT NULL,                            -- access mode (0 = local, 1 = central, 2 = shared)
  tp_rd_timed_ovr       SMALLINT NOT NULL,                            -- 1 = cardholder override enabled
  tp_strike_stat_enb    SMALLINT,                                     -- 1 = door strike messages enabled
  tp_output_stat_enb    SMALLINT,                                     -- 1 = output status messages enabled
  tp_interlock_enable   SMALLINT NOT NULL,                            -- 1 = allow action interlocks for this terminal
  tp_rd_override        SMALLINT NOT NULL,                            -- 1 = reader override enabled
  tp_cmd_seclevel       SMALLINT NOT NULL,                            -- 0 = not used. Last commanded security level value
  site                  CHAR(32) NOT NULL,                            -- Site Name
  tp_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  tp_checksum           INTEGER NOT NULL,                             -- Record checksum
  PRIMARY KEY           ( tp_partition, tp_term_id )
);
  CREATE INDEX              tp_public ON terminal(tp_public);
```

```
CREATE INDEX            tp_panel_id ON terminal ( tp_panel_id );
CREATE UNIQUE INDEX     tp_panel_index ON terminal ( tp_panel_id, tp_term_index );
CREATE UNIQUE INDEX     tp_site_name ON terminal (site, tp_term_name);
```

CREATE TABLE terminalstatus

```
(
  ts_term_id            INTEGER REFERENCES terminal(tp_term_id) NOT FOR REPLICATION,   -- unique terminal ID
  ts_rdr_status         SMALLINT,                                                     -- current reader status 0-2 (set by comms)
  ts_rdr_timestamp      DATETIME,                                                     -- current reader status time (set by comms)
  ts_in_status          SMALLINT,                                                     -- current input status 0-2 (set by comms)
  ts_in_timestamp       DATETIME,                                                     -- current input status time (set by comms)
  ts_out_status         SMALLINT,                                                     -- current output status 0-2 (set by comms)
  ts_out_timestamp      DATETIME,                                                     -- current output status time (set by comms)
  ts_read_seclevel      SMALLINT NOT NULL,                                            -- last read security level value
  site                  CHAR(32) NOT NULL,                                            -- Site Name
  ts_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                  -- Global unique record identifier
  ts_checksum           INTEGER NOT NULL,                                             -- Record checksum
  ts_details_timestamp  DATETIME NULL,
  ts_details1           VARCHAR(64) NULL,
  ts_details2           VARCHAR(64) NULL,
  ts_details3           VARCHAR(64) NULL,
  ts_details4           VARCHAR(64) NULL,
  ts_details5           VARCHAR(64) NULL,
  ts_details6           VARCHAR(64) NULL,
  ts_details7           VARCHAR(64) NULL,
  ts_details8           VARCHAR(64) NULL,
  PRIMARY KEY           (ts_term_id)
);
```

CREATE TABLE terminal_assaabloy

```
(
  [taa_id]                  INTEGER REFERENCES terminal(tp_term_id) PRIMARY KEY NOT NULL,
  [taa_public]              [smallint] NULL,
  [taa_query_string]        [char](64) NULL,
  [taa_type]                [varchar](64) NOT NULL,
  [taa_override_type]       [smallint] NULL,
  [taa_extended_access_time [smallint] NULL,
  [taa_pin_type]            [smallint] NULL,
  [taa_connect_interval]    [int] NOT NULL,
  [site]                    [char](32) NOT NULL,
  [taa_guid]                [uniqueidentifier] ROWGUIDCOL DEFAULT newid() NOT NULL,
  [taa_checksum]            [int] NOT NULL,
  [taa_wireless]            [smallint] NULL
 )
```

CREATE TABLE terminal_assaabloy_save

```
(
  [taa_id]                  [int] NOT NULL,
  [taa_public]              [smallint] NULL,
  [taa_query_string]        [char](64) NULL,
  [taa_type]                [varchar](64) NOT NULL,
  [taa_override_type]       [smallint] NULL,
  [taa_extended_access_time [smallint] NULL,
  [taa_pin_type]            [smallint] NULL,
  [taa_connect_interval]    [int] NOT NULL,
  [site]                    [char](32) NOT NULL,
  [taa_guid]                [uniqueidentifier] NOT NULL,
  [taa_checksum]            [int] NOT NULL,
  upd_guid                  uniqueidentifier NOT NULL,      -- Guid for update operation
  upd_mode                  SMALLINT NOT NULL,              -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp             DATETIME NOT NULL,              -- Date / Time of the update
```

```
  upd_checksum          INTEGER NOT NULL,                          -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                         -- Global unique record identifier
  [taa_wireless]        [smallint] NULL,
  PRIMARY KEY           (taa_id, upd_mode desc, upd_guid )
)
```

CREATE TABLE terminal_ck

```
(
  tp_id                 INTEGER REFERENCES terminal(tp_term_id)
                          NOT FOR REPLICATION,   -- Unique database ID of terminal
  tp_alarm_shunt        SMALLINT,              -- 1 = alarm shunt for aux access enabled
  tp_anti_tailgate      SMALLINT,              -- 1 = anti-tailgate enabled
  tp_fac_code_access    SMALLINT,              -- 1 = facility code only access enabled
  tp_moment_access      SMALLINT,              -- 1 = momentary aux access enabled
  tp_stand_alone_pin    SMALLINT,              -- 1 = pin required when offline enabled
  tp_deny_access        SMALLINT,              -- (legacy panels only)
  tp_search_card        SMALLINT,              -- (legacy panels only)
  tp_pin_after_badge    SMALLINT,              -- 1 = allow pin after badge enabled
  tp_warn_time          SMALLINT,              -- cardholder override time in mins (0-10)
  tp_warn_op_grp        INTEGER,               -- output group ID for cardholder override warning
  tp_link1_output1      SMALLINT NULL,         -- (reserved for D620 I/O linking)
  tp_link2_output2      SMALLINT NULL,         -- (reserved for D620 I/O linking)
  tp_link3_output3      SMALLINT NULL,         -- (reserved for D620 I/O linking)
  tp_link4_output4      SMALLINT NULL,         -- (reserved for D620 I/O linking)
  tp_io_linking         SMALLINT,              -- (reserved for D620 I/O linking)
  tp_alm_relay_mode     SMALLINT,              -- (reserved for legacy panels)
  tp_alm_rlay_to_act    SMALLINT,              -- (reserved for legacy panels)
  tp_key_switch_mode    SMALLINT,              -- (reserved for legacy panels)
  tp_fail_mode          SMALLINT,              -- 1 = fail secure enabled (legacy panels only)
  tp_fac_code_0         INTEGER NULL,          -- facility code 1
  tp_fac_code_1         INTEGER NULL,          -- facility code 2
  tp_fac_code_2         INTEGER NULL,          -- facility code 3
  tp_fac_code_3         INTEGER NULL,          -- facility code 4
  tp_fac_code_4         INTEGER NULL,          -- facility code 5
  tp_fac_code_5         INTEGER NULL,          -- facility code 6
  tp_fac_code_6         INTEGER NULL,          -- facility code 7
  tp_fac_code_7         INTEGER NULL,          -- facility code 8
  tp_fac_code_8         INTEGER NULL,          -- facility code 9
  tp_fac_code_9         INTEGER NULL,          -- facility code 10
  tp_fac_code_10        INTEGER NULL,          -- facility code 11
  tp_fac_code_11        INTEGER NULL,          -- facility code 12
  tp_fac_mask_0         INTEGER NULL,          -- facility code 1 card type
  tp_fac_mask_1         INTEGER NULL,          -- facility code 2 card type
  tp_fac_mask_2         INTEGER NULL,          -- facility code 3 card type
  tp_fac_mask_3         INTEGER NULL,          -- facility code 4 card type
  tp_fac_mask_4         INTEGER NULL,          -- facility code 5 card type
  tp_fac_mask_5         INTEGER NULL,          -- facility code 6 card type
  tp_fac_mask_6         INTEGER NULL,          -- facility code 7 card type
  tp_fac_mask_7         INTEGER NULL,          -- facility code 8 card type
  tp_fac_mask_8         INTEGER NULL,          -- facility code 9 card type
  tp_fac_mask_9         INTEGER NULL,          -- facility code 10 card type
  tp_fac_mask_10        INTEGER NULL,          -- facility code 11 card type
  tp_fac_mask_11        INTEGER NULL,          -- facility code 12 card type
  tp_alarm_debounce     SMALLINT,              -- alarm debounce time in msecs (20-80)
  tp_reverse            SMALLINT,              -- 1 = reverse reading enabled
  tp_reverse_duress     SMALLINT,              -- 1 = reverse swipe duress enabled
  tp_card_wiegand       TINYINT NULL,          -- 1 = wiegand cards enabled
  tp_card_encrypted     TINYINT NULL,          -- 1 = ncrypt cards enabled
  tp_card_magstripe     TINYINT NULL,          -- 1 = mag stripe cards enabled
  tp_card_binary        TINYINT NULL,          -- 1 = binary BaFe cards enabled
  tp_card_binary_par    TINYINT NULL,          -- 1 = binary BaFe parity cards enabled
  tp_card_pin           TINYINT NULL,          -- 1 = pin code enabled
```

```
    tp_card_id                TINYINT NULL,                                  -- 1 = badge id enabled
    tp_card_pin_id            TINYINT NULL,                                  -- 1 = pin code and badge id enabled
    tp_card_bcd               TINYINT NULL,                                  -- 1 = BCD BaFe cards enabled
    tp_card_bcd_par           TINYINT NULL,                                  -- 1 = BCD BaFe parity cards enabled
    tp_card_prox              TINYINT NULL,                                  -- 1 = Eyecam, Prox, Indala cards enabled
    tp_card_sensor_fwd        TINYINT NULL,                                  -- 1 = 26 bit sensor forward cards enabled
    tp_card_sensor_rev        TINYINT NULL,                                  -- 1 = 26 bit sensor reverse cards enabled
    tp_card_motorola          TINYINT NULL,                                  -- 1 = 32 bit Motorola cards enabled
    tp_card_custom            TINYINT NULL,                                  -- 1 = custom cards enabled
    tp_floor_sel_to           SMALLINT,                                      -- D600E floor select timeout (0 - 25 sec)
    tp_floor_acc_tmr          SMALLINT,                                      -- D600E floor access timer
    tp_card_type              SMALLINT,                                      -- Reader card type (non CK720 panels)
    tp_grant_on_open_only     SMALLINT NOT NULL,                             -- Send access grant message only if door is opened
                                                                                (v2.0 and higher)
    tp_door_open_warn_time    SMALLINT NOT NULL,                             -- Door open pre-warning time in seconds (0-255)
    tp_door_open_warn_op_grp  INTEGER NOT NULL,                              -- output group ID for door open pre-warning
    tp_air_crew_pin           SMALLINT,                                      -- D600-AP air crew pin P flag in prot-b dnld spec.
    tp_relock_on_open         TINYINT,                                       -- 1 = relock on door open enadled
    tp_timed_shunt            TINYINT,                                       -- 1 = timed shunt, 0 = timed override
    tp_star_feature           TINYINT,                                       -- 1 = star feature enabled
    tp_override_time          SMALLINT NOT NULL,                             -- override time with units min.
    tp_assisted_access        SMALLINT,                                      -- 2 = always, 1 = assisted A, 0 = never
    tp_assisted_access_time   SMALLINT,                                      -- assisted time with units sec, range 0-120
    tp_ada_relay_connector    SMALLINT,                                      -- 2 = green, 1 = shunt, 0 = none
    tp_ada_relay_time         SMALLINT,                                      -- ada relay time with units sec, range 0-120
    tp_ada_relay_delay        SMALLINT,                                      -- ada relay delay time with units 100ms, range 0-3000
    tp_pin_plus_1_duress      TINYINT,                                       -- 1 = pin plus 1 feature enadled
    tp_card_ccmag_stripe      TINYINT,                                       -- 1 = CCMAG stripe enadled
    tp_card_27_apd            TINYINT,                                       -- 1 = 27 bit APD enadled
    tp_card_hid_corp_1000     TINYINT,                                       -- 1 = hid corporate 1000 enadled
    tp_card_custom_1          TINYINT,                                       -- 1 = custom format 1 enadled
    tp_card_custom_2          TINYINT,                                       -- 1 = custom format 2 enadled
    tp_card_custom_3          TINYINT,                                       -- 1 = custom format 3 enadled
    tp_card_custom_4          TINYINT,                                       -- 1 = custom format 4 enadled
    tp_card_custom_5          TINYINT,                                       -- 1 = custom format 5 enadled
    tp_card_custom_6          TINYINT,                                       -- 1 = custom format 6 enadled
    tp_card_custom_7          TINYINT,                                       -- 1 = custom format 7 enadled
    tp_card_custom_8          TINYINT,                                       -- 1 = custom format 8 enadled
    tp_shunt_unit             TINYINT,                                       -- Used by D600AP with extened shunt enabled.
                                                                                1 = Minute, 0 = Seconds
    site                      CHAR(32) NOT NULL,                             -- Site Name
    tp_guid                   uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
    tp_checksum               INTEGER NOT NULL,                              -- Record checksum
    tp_nman_people            INTEGER NOT NULL,
    tp_nman_period            INTEGER NOT NULL,
    tp_ovrd_thrshld           INTEGER NOT NULL,
    tp_shunt_warn_auto_off    SMALLINT NOT NULL,
    tp_warn_auto_off          SMALLINT NOT NULL,
    tp_no_green_on_aux        SMALLINT NOT NULL,
    tp_bqt_lcd_reader         SMALLINT NOT NULL,
    tp_deny_if_door_open      SMALLINT NOT NULL,
    tp_visitor_escort_mode    SMALLINT NOT NULL,
    tp_rdr_module_type        SMALLINT NOT NULL,
    tp_rdr_module_addr        SMALLINT NOT NULL,
    tp_rdr_module_index       SMALLINT NOT NULL,
    tp_io_module_type         SMALLINT NOT NULL,
    tp_io_module_addr         SMALLINT NOT NULL,
    tp_io_module_index        SMALLINT NOT NULL,
    PRIMARY KEY               (tp_id)
);
```

CREATE TABLE terminal_ck_save

```
(
  tp_id                 INTEGER NOT NULL,   -- Unique database ID of terminal
  tp_alarm_shunt        SMALLINT,           -- 1 = alarm shunt for aux access enabled
  tp_anti_tailgate      SMALLINT,           -- 1 = anti-tailgate enabled
  tp_fac_code_access    SMALLINT,           -- 1 = facility code only access enabled
  tp_moment_access      SMALLINT,           -- 1 = momentary aux access enabled
  tp_stand_alone_pin    SMALLINT,           -- 1 = pin required when offline enabled
  tp_deny_access        SMALLINT,           -- (legacy panels only)
  tp_search_card        SMALLINT,           -- (legacy panels only)
  tp_pin_after_badge    SMALLINT,           -- 1 = allow pin after badge enabled
  tp_warn_time          SMALLINT,           -- cardholder override time in mins (0-10)
  tp_warn_op_grp        INTEGER,            -- output group ID for cardholder override warning
  tp_link1_output1      SMALLINT NULL,      -- (reserved for D620 I/O linking)
  tp_link2_output2      SMALLINT NULL,      -- (reserved for D620 I/O linking)
  tp_link3_output3      SMALLINT NULL,      -- (reserved for D620 I/O linking)
  tp_link4_output4      SMALLINT NULL,      -- (reserved for D620 I/O linking)
  tp_io_linking         SMALLINT,           -- (reserved for D620 I/O linking)
  tp_alm_relay_mode     SMALLINT,           -- (reserved for legacy panels)
  tp_alm_rlay_to_act    SMALLINT,           -- (reserved for legacy panels)
  tp_key_switch_mode    SMALLINT,           -- (reserved for legacy panels)
  tp_fail_mode          SMALLINT,           -- 1 = fail secure enabled (legacy panels only)
  tp_fac_code_0         INTEGER NULL,       -- facility code 1
  tp_fac_code_1         INTEGER NULL,       -- facility code 2
  tp_fac_code_2         INTEGER NULL,       -- facility code 3
  tp_fac_code_3         INTEGER NULL,       -- facility code 4
  tp_fac_code_4         INTEGER NULL,       -- facility code 5
  tp_fac_code_5         INTEGER NULL,       -- facility code 6
  tp_fac_code_6         INTEGER NULL,       -- facility code 7
  tp_fac_code_7         INTEGER NULL,       -- facility code 8
  tp_fac_code_8         INTEGER NULL,       -- facility code 9
  tp_fac_code_9         INTEGER NULL,       -- facility code 10
  tp_fac_code_10        INTEGER NULL,       -- facility code 11
  tp_fac_code_11        INTEGER NULL,       -- facility code 12
  tp_fac_mask_0         INTEGER NULL,       -- facility code 1 card type
  tp_fac_mask_1         INTEGER NULL,       -- facility code 2 card type
  tp_fac_mask_2         INTEGER NULL,       -- facility code 3 card type
  tp_fac_mask_3         INTEGER NULL,       -- facility code 4 card type
  tp_fac_mask_4         INTEGER NULL,       -- facility code 5 card type
  tp_fac_mask_5         INTEGER NULL,       -- facility code 6 card type
  tp_fac_mask_6         INTEGER NULL,       -- facility code 7 card type
  tp_fac_mask_7         INTEGER NULL,       -- facility code 8 card type
  tp_fac_mask_8         INTEGER NULL,       -- facility code 9 card type
  tp_fac_mask_9         INTEGER NULL,       -- facility code 10 card type
  tp_fac_mask_10        INTEGER NULL,       -- facility code 11 card type
  tp_fac_mask_11        INTEGER NULL,       -- facility code 12 card type
  tp_alarm_debounce     SMALLINT,           -- alarm debounce time in msecs (20-80)
  tp_reverse            SMALLINT,           -- 1 = reverse reading enabled
  tp_reverse_duress     SMALLINT,           -- 1 = reverse swipe duress enabled
  tp_card_wiegand       TINYINT NULL,       -- 1 = wiegand cards enabled
  tp_card_encrypted     TINYINT NULL,       -- 1 = ncrypt cards enabled
  tp_card_magstripe     TINYINT NULL,       -- 1 = mag stripe cards enabled
  tp_card_binary        TINYINT NULL,       -- 1 = binary BaFe cards enabled
  tp_card_binary_par    TINYINT NULL,       -- 1 = binary BaFe parity cards enabled
  tp_card_pin           TINYINT NULL,       -- 1 = pin code enabled
  tp_card_id            TINYINT NULL,       -- 1 = badge id enabled
  tp_card_pin_id        TINYINT NULL,       -- 1 = pin code and badge id enabled
  tp_card_bcd           TINYINT NULL,       -- 1 = BCD BaFe cards enabled
  tp_card_bcd_par       TINYINT NULL,       -- 1 = BCD BaFe parity cards enabled
  tp_card_prox          TINYINT NULL,       -- 1 = Eyecam, Prox, Indala cards enabled
  tp_card_sensor_fwd    TINYINT NULL,       -- 1 = 26 bit sensor forward cards enabled
  tp_card_sensor_rev    TINYINT NULL,       -- 1 = 26 bit sensor reverse cards enabled
  tp_card_motorola      TINYINT NULL,       -- 1 = 32 bit Motorola cards enabled
  tp_card_custom        TINYINT NULL,       -- 1 = custom cards enabled
```

```
  tp_floor_sel_to           SMALLINT,                          -- D600E floor select timeout (0 - 25 sec)
  tp_floor_acc_tmr          SMALLINT,                          -- D600E floor access timer
  tp_card_type              SMALLINT,                          -- Reader card type (non CK720 panels)
  tp_grant_on_open_only     SMALLINT NOT NULL,                 -- Send access grant message only if door is opened
                                                                  (v2.0 and higher)
  tp_door_open_warn_time    SMALLINT NOT NULL,                 -- Door open pre-warning time in seconds (0-255)
  tp_door_open_warn_op_grp  INTEGER NOT NULL,                  -- output group ID for door open pre-warning
  --tp_security_level       SMALLINT,                          -- security level 0-99
  tp_air_crew_pin           SMALLINT,                          -- D600-AP air crew pin P flag in prot-b dnld spec.
  tp_relock_on_open         TINYINT,                           -- 1 = relock on door open enadled
  tp_timed_shunt            TINYINT,                           -- 1 = timed shunt, 0 = timed override
  tp_star_feature           TINYINT,                           -- 1 = star feature enabled
  tp_override_time          SMALLINT NOT NULL,                 -- override time with units min.
  tp_assisted_access        SMALLINT,                          -- 2 = always, 1 = assisted A, 0 = never
  tp_assisted_access_time   SMALLINT,                          -- assisted time with units sec, range 0-120
  tp_ada_relay_connector    SMALLINT,                          -- 2 = green, 1 = shunt, 0 = none
  tp_ada_relay_time         SMALLINT,                          -- ada relay time with units sec, range 0-120
  tp_ada_relay_delay        SMALLINT,                          -- ada relay delay time with units 100ms, range 0-3000
  tp_pin_plus_1_duress      TINYINT,                           -- 1 = pin plus 1 feature enadled
  tp_card_ccmag_stripe      TINYINT,                           -- 1 = CCMAG stripe enadled
  tp_card_27_apd            TINYINT,                           -- 1 = 27 bit APD enadled
  tp_card_hid_corp_1000     TINYINT,                           -- 1 = hid corporate 1000 enadled
  tp_card_custom_1          TINYINT,                           -- 1 = custom format 1 enadled
  tp_card_custom_2          TINYINT,                           -- 1 = custom format 2 enadled
  tp_card_custom_3          TINYINT,                           -- 1 = custom format 3 enadled
  tp_card_custom_4          TINYINT,                           -- 1 = custom format 4 enadled
  tp_card_custom_5          TINYINT,                           -- 1 = custom format 5 enadled
  tp_card_custom_6          TINYINT,                           -- 1 = custom format 6 enadled
  tp_card_custom_7          TINYINT,                           -- 1 = custom format 7 enadled
  tp_card_custom_8          TINYINT,                           -- 1 = custom format 8 enadled
  tp_shunt_unit             TINYINT,                           -- Used by D600AP with extened shunt enabled. 1 = Minute,
                                                                  0 = Seconds
  site                      CHAR(32) NOT NULL,                 -- Site Name
  tp_guid                   uniqueidentifier NOT NULL,         -- Global unique record identifier
  tp_checksum               INTEGER NOT NULL,                  -- Record checksum
  upd_guid                  uniqueidentifier NOT NULL,         -- Guid for update operation
  upd_mode                  SMALLINT NOT NULL,                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp             DATETIME NOT NULL,                 -- Date / Time of the update
  upd_checksum              INTEGER NOT NULL,                  -- Checksum
  upd_rowguid               uniqueidentifier ROWGUIDCOL DEFAULT newid()
                              UNIQUE NOT NULL,                 -- Global unique record identifier
  tp_nman_people            INTEGER NOT NULL,
  tp_nman_period            INTEGER NOT NULL,
  tp_ovrd_thrshld           INTEGER NOT NULL,
  tp_shunt_warn_auto_off    SMALLINT,
  tp_warn_auto_off          SMALLINT,
  tp_no_green_on_aux        SMALLINT,
  tp_bqt_lcd_reader         SMALLINT,
  tp_deny_if_door_open      SMALLINT,
  tp_visitor_escort_mode    SMALLINT,
  tp_rdr_module_type        SMALLINT,
  tp_rdr_module_addr        SMALLINT,
  tp_rdr_module_index       SMALLINT,
  tp_io_module_type         SMALLINT,
  tp_io_module_addr         SMALLINT,
  tp_io_module_index        SMALLINT,
  PRIMARY KEY               (tp_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE terminal_hid

```
(
  thid_id                   INTEGER REFERENCES terminal(tp_term_id)
                              NOT FOR REPLICATION,             -- Unique database ID of panel
```

```
  thid_card_format              VARCHAR(4096) NULL,              -- Card format
  thid_extended_access_time     SMALLINT NOT NULL,               -- Extended access time in seconds 0 - 65535
  thid_rex_input_unlocks_door   SMALLINT NOT NULL,               -- 1 = rex unlocks door, 0 = rex does not unlock door
  thid_annunciate_propped_door  SMALLINT NOT NULL,               -- 1 = annunicate locally, 0 = do not annunicate
  thid_anti_passback_action     SMALLINT NOT NULL,               -- 0 = No action, 1 = Notify P2000,
                                                                    2 = Deny access and Notify P2000
  thid_aux_tracks_tamper        SMALLINT NOT NULL,               -- 1 = Aux Relay follows Tamper state, 0 = does not follow
  thid_pin_access_method        SMALLINT NOT NULL,               -- 1 = Card only, 2 = Card & PIN, 3 = Card ID,
                                                                    4 = Card or Card ID
  thid_pin_max_entry_time       SMALLINT NOT NULL,               -- 0 - 60 seconds, maximum time to enter pin
  thid_pin_max_attempts         SMALLINT NOT NULL,               -- 1 - 10 maximum number of pin entry attempts
  thid_pin_lockout_time         SMALLINT NOT NULL,               -- 0 - 99 seconds, lockout time on failed attempt
  thid_user_linker_rules_used   SMALLINT NOT NULL,               -- Are IO Linker rules added by field? (Note: No UI adds to
                                                                    this column)
  thid_user_linker_rules        VARCHAR(4096) NULL,              -- IO Linker rules added from this field (Note: No UI adds to
                                                                    this column)
  site                          CHAR(32) NOT NULL,               -- Site Name
  thid_guid                     uniqueidentifier ROWGUIDCOL UNIQUE
                                   NOT NULL,                     -- Global unique record identifier
  thid_checksum                 INTEGER NOT NULL,                -- Record checksum
  PRIMARY KEY                   (thid_id)
);
```

CREATE TABLE terminal_hid_save

```
(
  thid_id                       INTEGER NOT NULL,                -- Unique database ID of terminal
  thid_card_format              VARCHAR(4096) NULL,              -- Card format
  thid_extended_access_time     SMALLINT NOT NULL,               -- Extended access time in seconds 0 - 65535
  thid_rex_input_unlocks_door   SMALLINT NOT NULL,               -- 1 = rex unlocks door, 0 = rex does not unlock door
  thid_annunciate_propped_door  SMALLINT NOT NULL,               -- 1 = annunicate locally, 0 = do not annunicate
  thid_anti_passback_action     SMALLINT NOT NULL,               -- 0 = No action, 1 = Notify P2000, 2 = Deny access and Notify P2000
  thid_aux_tracks_tamper        SMALLINT NOT NULL,               -- 1 = Aux Relay follows Tamper state, 0 = does not follow
  thid_pin_access_method        SMALLINT NOT NULL,               -- 1 = Card only, 2 = Card & PIN, 3 = Card ID, 4 = Card or Card ID
  thid_pin_max_entry_time       SMALLINT NOT NULL,               -- 0 - 60 seconds, maximum time to enter pin
  thid_pin_max_attempts         SMALLINT NOT NULL,               -- 1 - 10 maximum number of pin entry attempts
  thid_pin_lockout_time         SMALLINT NOT NULL,               -- 0 - 99 seconds, lockout time on failed attempt
  thid_user_linker_rules_used   SMALLINT NOT NULL,               -- Are IO Linker rules added by field? (Note: No UI adds to this
                                                                    column)
  thid_user_linker_rules        VARCHAR(4096) NULL,              -- IO Linker rules added by field (Note: No UI adds to this column)
  site                          CHAR(32) NOT NULL,               -- Site Name
  thid_guid                     uniqueidentifier NOT NULL,       -- Global unique record identifier
  thid_checksum                 INTEGER NOT NULL,                -- Record checksum
  upd_guid                      uniqueidentifier NOT NULL,       -- Guid for update operation
  upd_mode                      SMALLINT NOT NULL,               -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                 DATETIME NOT NULL,               -- Date / Time of the update
  upd_checksum                  INTEGER NOT NULL,                -- Checksum
  upd_rowguid                   uniqueidentifier ROWGUIDCOL DEFAULT
                                   newid() UNIQUE NOT NULL,      -- Global unique record identifier
  PRIMARY KEY                   (thid_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE terminal_isonas

```
(
  tisonas_id                    INTEGER REFERENCES terminal(tp_term_id)
                                   NOT FOR REPLICATION,          -- Unique database ID of panel
  tisonas_unauth_activate_ttl2  TINYINT NOT NULL,                -- 1 = Active TTL2 when door forced or held 0 = do nothing
  tisonas_rex_unlock_door       TINYINT NOT NULL,                -- 1 = On REX unlock door 0 = On REX Activate TTL2
  tisonas_rex_beep_unlock       TINYINT NOT NULL,                -- 1 = On REX unlock beep 0 = On REX unlock don't beep
  tisonas_aux_activate_ttl1     TINYINT NOT NULL,                -- 1 = On AUX activate TTL1 0 = On AUX don't activate TTL1
  tisonas_aux_activate_ttl2     TINYINT NOT NULL,                -- 1 = On AUX activate TTL2 0 = On AUX don't activate TTL2
  tisonas_aux_activate_relay    TINYINT NOT NULL,                -- 1 = On AUX activate relay 0 = On AUX don't activate relay
  tisonas_tamper_beep           TINYINT NOT NULL,                -- 1 = On Tamper beep 0 = On Tamper don't beep
```

```
  tisonas_tamper_activate_ttl1    TINYINT NOT NULL,                    -- 1 = On Tamper activate TTL1 0 = On Tamper don't activate TTL1
  tisonas_card_number_source      TINYINT NOT NULL,                    -- 1 = Badge, 2 = Keypad, 3 = Both badge and keypad
  tisonas_card_format             VARCHAR(4096) NULL,                  -- Card format
  tisonas_card_bits_to_use        CHAR(24) NULL,                       -- Bit mask of which 32 out of 96 bits to use
  site                            CHAR(32) NOT NULL,                   -- Site Name
  tisonas_guid                    uniqueidentifier ROWGUIDCOL UNIQUE
                                    NOT NULL,                          -- Global unique record identifier
  tisonas_checksum                INTEGER NOT NULL,                    -- Record checksum
  PRIMARY KEY                     (tisonas_id)
);
```

CREATE TABLE terminal_isonas_save

```
(
  tisonas_id                      INTEGER NOT NULL,                    -- Unique database ID of terminal
  tisonas_unauth_activate_ttl2    TINYINT NOT NULL,                    -- 1 = Active TTL2 when door forced or held 0 = do nothing
  tisonas_rex_unlock_door         TINYINT NOT NULL,                    -- 1 = On REX unlock door 0 = On REX Activate TTL2
  tisonas_rex_beep_unlock         TINYINT NOT NULL,                    -- 1 = On REX unlock beep 0 = On REX unlock don't beep
  tisonas_aux_activate_ttl1       TINYINT NOT NULL,                    -- 1 = On AUX activate TTL1 0 = On AUX don't activate TTL1
  tisonas_aux_activate_ttl2       TINYINT NOT NULL,                    -- 1 = On AUX activate TTL2 0 = On AUX don't activate TTL2
  tisonas_aux_activate_relay      TINYINT NOT NULL,                    -- 1 = On AUX activate relay 0 = On AUX don't activate relay
  tisonas_tamper_beep             TINYINT NOT NULL,                    -- 1 = On Tamper beep 0 = On Tamper don't beep
  tisonas_tamper_activate_ttl1    TINYINT NOT NULL,                    -- 1 = On Tamper activate TTL1 0 = On Tamper don'tactivate TTL1
  tisonas_card_number_source      TINYINT NOT NULL,                    -- 1 = Badge, 2 = Keypad, 3 = Both badge and keypad
  tisonas_card_format             VARCHAR(4096) NULL,                  -- Card format
  tisonas_card_bits_to_use        CHAR(24) NULL,                       -- Bit mask of which 32 out of 96 bits to use
  site                            CHAR(32) NOT NULL,                   -- Site Name
  tisonas_guid                    uniqueidentifier NOT NULL,           -- Global unique record identifier
  tisonas_checksum                INTEGER NOT NULL,                    -- Record checksum
  upd_guid                        uniqueidentifier NOT NULL,           -- Guid for update operation
  upd_mode                        SMALLINT NOT NULL,                   -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                   DATETIME NOT NULL,                   -- Date / Time of the update
  upd_checksum                    INTEGER NOT NULL,                    -- Checksum
  upd_rowguid                     uniqueidentifier ROWGUIDCOL DEFAULT
                                    newid() UNIQUE NOT NULL,           -- Global unique record identifier
  PRIMARY KEY                     (tisonas_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE terminal_osi

```
(
  tosi_id                 INTEGER REFERENCES terminal(tp_term_id) NOT FOR REPLICATION,   -- Unique database ID of panel
  tosi_mac                CHAR(16) UNIQUE NOT NULL,                                     -- mac address of terminal
  tosi_door_sensors       SMALLINT NOT NULL,                                            -- 1 = reader has door sensors
  tosi_update_interval    INTEGER NOT NULL,
  site                    CHAR(32) NOT NULL,                                            -- Site Name
  tosi_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                  -- Global unique record identifier
  tosi_checksum           INTEGER NOT NULL,                                             -- Record checksum
  tosi_channels           INTEGER NOT NULL,                                             -- Record channels
  PRIMARY KEY             (tosi_id)
);
```

CREATE TABLE terminal_osi_save

```
(
  tosi_id                 INTEGER NOT NULL,                                             -- Unique database ID of terminal
  tosi_mac                CHAR(16) NOT NULL,                                            -- mac address of terminal
  tosi_door_sensors       SMALLINT NOT NULL,                                            -- 1 = reader has door sensors
  tosi_update_interval    INTEGER NOT NULL,
  site                    CHAR(32) NOT NULL,                                            -- Site Name
  tosi_guid               uniqueidentifier NOT NULL,                                    -- Global unique record identifier
  tosi_checksum           INTEGER NOT NULL,                                             -- Record checksum
  tosi_channels           INTEGER ,                                                     -- Record channels
  upd_guid                uniqueidentifier NOT NULL,                                    -- Guid for update operation
```

```
  upd_mode              SMALLINT NOT NULL,                                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                          -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                           -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY           (tosi_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE terminal_p900

```
(
  tpp_id                    INTEGER REFERENCES terminal(tp_term_id)
                              NOT FOR REPLICATION,      -- Unique database ID of terminal
  tpp_sys_addr              CHAR(8),                  -- address
  tpp_reader_interface_num  SMALLINT,                 -- reader interface number
  tpp_door_relay_type       SMALLINT,                 -- door relay time type
  tpp_door_control          SMALLINT,                 -- door control 0: disable 1: enable
  tpp_egress                SMALLINT,                 -- egress 0: disable 1: enable 2: report
  tpp_door_monitor_enable   SMALLINT,                 -- door monitoring enable, 0: disable 1: enable
  tpp_door_forced_alarm     SMALLINT,                 -- door forced alarm, 0: disable 1: enable
  tpp_door_forced_warning   SMALLINT,                 -- door forced warning, 0: disable 1: enable
  tpp_door_open_alarm       SMALLINT,                 -- door open alarm, 0: disable 1: enable
  tpp_door_open_alarm_delay SMALLINT,                 -- door open alarm delay 1-255(second)
  tpp_door_open_warning     SMALLINT,                 -- door open warning, 0: disable 1: enable
  tpp_door_open_warning_delay SMALLINT,               -- door open warning delay 1-255(second)
  tpp_global_apb_status     SMALLINT,                 -- global anti-passback status 0: disable 1: entry based
                                                        2: exit based 3: Shunt
  tpp_local_apb_status      SMALLINT,                 -- local anti-passback status 0: disable 1: in reader 2: out reader
  tpp_enter_zone            INTEGER,                  -- entering zone
  tpp_exit_zone             INTEGER,                  -- exiting zone
  tpp_shunt_terminal        INTEGER REFERENCES terminal(tp_term_id)
                              NOT FOR REPLICATION,    -- shunt terminal 1-63
  tpp_timestamp             DATETIME,                 -- time stamp
  tpp_time_record           SMALLINT,                 -- time record mode
  tpp_pinpad_mode           SMALLINT,                 -- pin pad mode 0: disable 1: type #1 2: type #2 3: type #3
  tpp_aux_input_tamper      SMALLINT,                 -- aux input tamper 0: disable 1: enable
  tpp_aux_input_monitor     SMALLINT,                 -- aux input Monitor 0: disable 1: enable
  tpp_aux_input_disable_tz  INTEGER NULL REFERENCES timezone(tz_id)
                              NOT FOR REPLICATION,    -- aux input disable time zone
  tpp_aux_input_desp        char(64),                 -- aux input description
  tpp_aux_output_control    SMALLINT,                 -- aux output control 0: disable 1: enable
  tpp_aux_op_under_pin      SMALLINT,                 -- aux output under pin 0: disable 1: enable
  tpp_aux_output_ctl_pin    char(4),                  -- aux output control pin number
  tpp_pin_check             SMALLINT,                 -- under pin control or not
  tpp_tz_check              SMALLINT,                 -- using tz or not
  tpp_aux_output_tz         INTEGER NULL REFERENCES timezone(tz_id)
                              NOT FOR REPLICATION,    -- aux output time zone
  tpp_aux_output_desp       char(64),                 -- aux output description
  tpp_trace_mode            SMALLINT,                 -- transaction trace mode 1: trace 2: seleted
  tpp_selected              SMALLINT NOT NULL,        -- selected or not
  tpp_tamper_monitoring_door SMALLINT NOT NULL,       -- tamper monitoring of door contact or not
  tpp_tamper_monitoring_egress SMALLINT NOT NULL,     -- tamper monitoring of egress contact or not
  tpp_clic_pin              char(4),                  -- clic pin number
  tpp_swipe_pin             SMALLINT NOT NULL,        -- swipe pin data entry or not
  site                      CHAR(32) NOT NULL,        -- Site Name
  tpp_guid                  uniqueidentifier ROWGUIDCOL UNIQUE
                              NOT NULL,               -- Global unique record identifier
  tpp_checksum              INTEGER NOT NULL,         -- Record checksum
  PRIMARY KEY               (tpp_id)
);
  CREATE INDEX              tpp_selected ON terminal_p900(tpp_selected);
```

CREATE TABLE terminal_p900_save

```
(
  tpp_id                        INTEGER NOT NULL,                  -- Unique database ID of terminal
  tpp_sys_addr                  CHAR(8),                           -- address
  tpp_reader_interface_num      SMALLINT,                          -- reader interface number
  tpp_door_relay_type           SMALLINT,                          -- door relay time type
  tpp_door_control              SMALLINT,                          -- door control 0: disable 1: enable
  tpp_egress                    SMALLINT,                          -- egress 0: disable 1: enable 2: report
  tpp_door_monitor_enable       SMALLINT,                          -- door monitoring enable, 0: disable 1: enable
  tpp_door_forced_alarm         SMALLINT,                          -- door forced alarm, 0: disable 1: enable
  tpp_door_forced_warning       SMALLINT,                          -- door forced warning, 0: disable 1: enable
  tpp_door_open_alarm           SMALLINT,                          -- door open alarm, 0: disable 1: enable
  tpp_door_open_alarm_delay     SMALLINT,                          -- door open alarm delay 1-255(second)
  tpp_door_open_warning         SMALLINT,                          -- door open warning, 0: disable 1: enable
  tpp_door_open_warning_delay   SMALLINT,                          -- door open warning delay 1-255(second)
  tpp_global_apb_status         SMALLINT,                          -- global anti-passback status 0: disable 1: entry based
                                                                      2: exit based 3: Shunt
  tpp_local_apb_status          SMALLINT,                          -- local anti-passback status 0: disable 1: in reader 2: out reader
  tpp_enter_zone                INTEGER,                           -- entering zone
  tpp_exit_zone                 INTEGER,                           -- exiting zone
  tpp_shunt_terminal            INTEGER,                           -- shunt terminal 1-63
  tpp_timestamp                 DATETIME,                          -- time stamp
  tpp_time_record               SMALLINT,                          -- time record mode
  tpp_pinpad_mode               SMALLINT,                          -- pin pad mode 0: disable 1: type #1 2: type #2 3: type #3
  tpp_aux_input_tamper          SMALLINT,                          -- aux input tamper 0: disable 1: enable
  tpp_aux_input_monitor         SMALLINT,                          -- aux input Monitor 0: disable 1: enable
  tpp_aux_input_disable_tz      INTEGER NULL,                      -- aux input disable time zone
  tpp_aux_input_desp            char(64),                          -- aux input description
  tpp_aux_output_control        SMALLINT,                          -- aux output control 0: disable 1: enable
  tpp_aux_op_under_pin          SMALLINT,                          -- aux output under pin 0: disable 1: enable
  tpp_aux_output_ctl_pin        char(4),                           -- aux output control pin number
  tpp_pin_check                 SMALLINT,                          -- under pin control or not
  tpp_tz_check                  SMALLINT,                          -- using tz or not
  tpp_aux_output_tz             INTEGER NULL,                      -- aux output time zone
  tpp_aux_output_desp           char(64),                          -- aux output description
  tpp_trace_mode                SMALLINT,                          -- transaction trace mode 1: trace 2: seleted
  tpp_selected                  SMALLINT NOT NULL,                 -- selected or not
  tpp_tamper_monitoring_door    SMALLINT NOT NULL,                 -- tamper monitoring of door contact or not
  tpp_tamper_monitoring_egress  SMALLINT NOT NULL,                 -- tamper monitoring of egress contact or not
  tpp_clic_pin                  char(4),                           -- clic pin number
  tpp_swipe_pin                 SMALLINT NOT NULL,                 -- swipe pin data entry or not
  site                          CHAR(32) NOT NULL,                 -- Site Name
  tpp_guid                      uniqueidentifier NOT NULL,         -- Global unique record identifier
  tpp_checksum                  INTEGER NOT NULL,                  -- Record checksum
  upd_guid                      uniqueidentifier NOT NULL,         -- Guid for update operation
  upd_mode                      SMALLINT NOT NULL,                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                 DATETIME NOT NULL,                 -- Date / Time of the update
  upd_checksum                  INTEGER NOT NULL,                  -- Checksum
  upd_rowguid                   uniqueidentifier ROWGUIDCOL DEFAULT
                                  newid() UNIQUE NOT NULL,         -- Global unique record identifier
  PRIMARY KEY                   (tpp_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE terminal_s321ip

```
(
  ts321ip_id                    INTEGER REFERENCES terminal(tp_term_id)
                                  NOT FOR REPLICATION,             -- Unique database ID of panel
  ts321ip_reader_mode_not_io    TINYINT NOT NULL,                  -- 1 = reader enabled 0 = reader disabled
  ts321ip_shunt_alarm_on_aux    TINYINT NOT NULL,                  -- 1 = shunt alarm on auxiliary access
  ts321ip_moment_access         TINYINT NOT NULL,                  -- 1 = momentary aux access enabled
  ts321ip_badge_id_allowed      TINYINT NOT NULL,                  -- 1 = badge number can be entered on key pad
  ts321ip_card_hid_1000         TINYINT NOT NULL,                  -- 1 = HID corp 1000
  ts321ip_card_ck_standard      TINYINT NOT NULL,                  -- 1 = Cardkey standard (aka Wiegand 34)
```

```
  ts321ip_card_ck_magstripe       TINYINT NOT NULL,                          -- 1 = cardkey mag stripe cards enabled
  ts321ip_card_26_bit_sensor      TINYINT NOT NULL,                          -- 1 = sensor 26 bit cards enabled
  ts321ip_card_raw_128_bit        TINYINT NOT NULL,                          -- 1 = raw 128 bit card format
  ts321ip_card_raw_128_bit_length TINYINT NOT NULL,                          -- 0 - 128 bit length of raw 128 bit card
  ts321ip_override_reset_level    INTEGER NOT NULL,                          -- 0 - 99 used with security level to reset overrides
  ts321ip_card_custom             TINYINT NOT NULL,                          -- 0 = no custom, 1 - 8 custom card format
  site                            CHAR(32) NOT NULL,                         -- Site Name
  ts321ip_guid                    uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  ts321ip_checksum                INTEGER NOT NULL,                          -- Record checksum
  PRIMARY KEY                     (ts321ip_id)
);
```

CREATE TABLE terminal_s321ip_save

```
(
  ts321ip_id                      INTEGER NOT NULL,                          -- Unique database ID of terminal
  ts321ip_reader_mode_not_io      TINYINT NOT NULL,                          -- 1 = reader enabled 0 = reader disabled
  ts321ip_shunt_alarm_on_aux      TINYINT NOT NULL,                          -- 1 = shunt alarm on auxiliary access
  ts321ip_moment_access           TINYINT NOT NULL,                          -- 1 = momentary aux access enabled
  ts321ip_badge_id_allowed        TINYINT NOT NULL,                          -- 1 = badge number can be entered on key pad
  ts321ip_card_hid_1000           TINYINT NOT NULL,                          -- 1 = HID corp 1000 enabled
  ts321ip_card_ck_standard        TINYINT NOT NULL,                          -- 1 = Cardkey standard (aka Wiegand 34)
  ts321ip_card_ck_magstripe       TINYINT NOT NULL,                          -- 1 = cardkey mag stripe cards enabled
  ts321ip_card_26_bit_sensor      TINYINT NOT NULL,                          -- 1 = 26 bit sensor cards enabled
  ts321ip_card_raw_128_bit        TINYINT NOT NULL,                          -- 1 = 32 bit raw card format
  ts321ip_card_raw_128_bit_length TINYINT NOT NULL,                          -- 0 - 128 bit length of raw 128 bit card
  ts321ip_override_reset_level    INTEGER NOT NULL,                          -- 0 - 99 used with security level to reset overrides
  ts321ip_card_custom             TINYINT NOT NULL,                          -- 0 = no custom, 1 - 8 custom card format
  site                            CHAR(32) NOT NULL,                         -- Site Name
  ts321ip_guid                    uniqueidentifier NOT NULL,                 -- Global unique record identifier
  ts321ip_checksum                INTEGER NOT NULL,                          -- Record checksum
  upd_guid                        uniqueidentifier NOT NULL,                 -- Guid for update operation
  upd_mode                        SMALLINT NOT NULL,                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                   DATETIME NOT NULL,                         -- Date / Time of the update
  upd_checksum                    INTEGER NOT NULL,                          -- Checksum
  upd_rowguid                     uniqueidentifier ROWGUIDCOL DEFAULT newid()
                                      UNIQUE NOT NULL,                       -- Global unique record identifier
  PRIMARY KEY                     (ts321ip_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE terminal_save

```
(
  tp_term_id          INTEGER NOT NULL,          -- unique terminal ID
  tp_term_name        varchar(32) NOT NULL,      -- name of terminal
  tp_model            SMALLINT NOT NULL,         -- terminal Model
  tp_term_type        SMALLINT NOT NULL,         -- terminal type
  tp_term_index       SMALLINT NOT NULL,         -- terminal index (0-15)
  tp_partition        INTEGER,                   -- ID of partition for this terminal
  tp_public           SMALLINT NOT NULL,         -- publication
  tp_panel_id         INTEGER  NOT NULL,         -- panel id
  tp_query_string     CHAR(64),                  -- query string enter by user
  tp_reader_flag      SMALLINT NOT NULL,         -- 1 = terminal is a reader terminal
  tp_input_flag       SMALLINT NOT NULL,         -- 1 = terminal is an input terminal
  tp_output_flag      SMALLINT NOT NULL,         -- 1 = terminal is an output terminal
  tp_enb_dis          SMALLINT NOT NULL,         -- 1 = terminal is enabled
  tp_term_tz          INTEGER  NULL,             -- timezone ID when enabled (always if NULL)
  tp_override_tz      INTEGER  NULL,             -- timezone ID when override (always if NULL)
  tp_pin_enable_flag  SMALLINT NOT NULL,         -- pin pad enable/disable
  tp_pin_enable_tz    INTEGER  NULL,             -- timezone ID when pin code suppressed (always if NULL)
  tp_pin_disable_tz   INTEGER  NULL,             -- pin pad disable time zone
  tp_access_time      SMALLINT NOT NULL,         -- access time in secs (0-25)
  tp_shunt_time       SMALLINT NOT NULL,         -- shunt time in secs (0-255)
  tp_manual_rdr       SMALLINT NOT NULL,         -- 1 = valid and unauthorized enabled
```

```
  tp_anti_passback      SMALLINT NOT NULL,                          -- 1 = anti-passback enabled
  tp_anti_pb_time       SMALLINT NOT NULL,                          -- anti-passback time in mins (0-60)
  tp_ntr_xit_time       SMALLINT NOT NULL,                          -- entry/exit time in secs (0-255)
  tp_soft_inxit         SMALLINT NOT NULL,                          -- 1 = soft entry/exit enabled
  tp_access_modes       SMALLINT NOT NULL,                          -- access mode (0 = local, 1 = central, 2 = shared)
  tp_rd_timed_ovr       SMALLINT NOT NULL,                          -- 1 = cardholder override enabled
  tp_strike_stat_enb    SMALLINT,                                   -- 1 = door strike messages enabled
  tp_output_stat_enb    SMALLINT,                                   -- 1 = output status messages enabled
  tp_interlock_enable   SMALLINT NOT NULL,                          -- 1 = allow action interlocks for this terminal
  tp_rd_override        SMALLINT NOT NULL,                          -- 1 = reader override enabled
  tp_cmd_seclevel       SMALLINT NOT NULL,                          -- 0 = not used. Last commanded security level value
  site                  CHAR(32) NOT NULL,                          -- Site Name
  tp_guid               uniqueidentifier NOT NULL,                  -- Global unique record identifier
  tp_checksum           INTEGER NOT NULL,                           -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                           -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                          -- Global unique record identifier
  PRIMARY KEY           ( tp_partition, tp_term_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE term_pin_number

```
(
  tpn_term_id           INTEGER REFERENCES terminal(tp_term_id) NOT FOR REPLICATION,   -- terminal ID
  tpn_pin_id            INTEGER REFERENCES pin_number(pn_id) NOT FOR REPLICATION,      -- pin ID
  site                  CHAR(32) NOT NULL,                                            -- Site Name
  tpn_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                  -- Global unique record identifier
  tpn_checksum          INTEGER NOT NULL,                                             -- Record checksum
  PRIMARY KEY           ( tpn_term_id, tpn_pin_id )
);
```

CREATE TABLE term_pin_number_save

```
(
  tpn_term_id           INTEGER,                                                      -- terminal ID
  tpn_pin_id            INTEGER,                                                      -- pin ID
  site                  CHAR(32) NOT NULL,                                            -- Site Name
  tpn_guid              uniqueidentifier NOT NULL,                                    -- Global unique record identifier
  tpn_checksum          INTEGER NOT NULL,                                             -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                                    -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                            -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                            -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                             -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,  -- Global unique record identifier
  PRIMARY KEY           ( tpn_term_id, tpn_pin_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE timeattendanceterminals

```
(
  tat_id                INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,   -- unique database ID
  terminal_name         CHAR(32) UNIQUE NOT NULL,                      -- Terminal Name
  entryexit             SMALLINT NOT NULL,                             -- 1 = entry, 0 = exit
  zone_name             CHAR(32) NOT NULL,                             -- Zone Name
  PRIMARY KEY           (tat_id)
);
```

CREATE TABLE timezone

```
(
  tz_partition          INTEGER REFERENCES partition(part_number)
                          NOT FOR REPLICATION,          -- partition ID to which this timezone belongs
  tz_public             SMALLINT,                       -- 1 = public, 0 = private
  tz_id                 INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,  -- unique ID for this timezone
  tz_name               CHAR(32) NOT NULL,              -- timezone name
  tz_init_stat_0        SMALLINT,                       -- 1 = monday initial status active
  tz_init_stat_1        SMALLINT,                       -- 1 = tuesday initial status active
  tz_init_stat_2        SMALLINT,                       -- 1 = wednesday initial status active
  tz_init_stat_3        SMALLINT,                       -- 1 = thursday initial status active
  tz_init_stat_4        SMALLINT,                       -- 1 = friday initial status active
  tz_init_stat_5        SMALLINT,                       -- 1 = saturday initial status active
  tz_init_stat_6        SMALLINT,                       -- 1 = sunday initial status active
  tz_init_stat_7        SMALLINT,                       -- 1 = holiday 1 initial status active
  tz_init_stat_8        SMALLINT,                       -- 1 = holiday 2 initial status active
  tz_init_stat_9        SMALLINT,                       -- 1 = holiday 3 initial status active
  tz_times_0_0          DATETIME,                       -- monday time 1
  tz_times_0_1          DATETIME,                       -- monday time 2
  tz_times_0_2          DATETIME,                       -- monday time 3
  tz_times_0_3          DATETIME,                       -- monday time 4
  tz_times_0_4          DATETIME,                       -- monday time 5
  tz_times_0_5          DATETIME,                       -- monday time 6
  tz_times_0_6          DATETIME,                       -- monday time 7
  tz_times_0_7          DATETIME,                       -- monday time 8
  tz_times_0_8          DATETIME,                       -- monday time 9
  tz_times_0_9          DATETIME,                       -- monday time 10
  tz_times_0_10         DATETIME,                       -- monday time 11
  tz_times_0_11         DATETIME,                       -- monday time 12
  tz_times_0_12         DATETIME,                       -- monday time 13
  tz_times_0_13         DATETIME,                       -- monday time 14
  tz_times_0_14         DATETIME,                       -- monday time 15
  tz_times_0_15         DATETIME,                       -- monday time 16
  tz_times_0_16         DATETIME,                       -- monday time 17
  tz_times_0_17         DATETIME,                       -- monday time 18
  tz_times_0_18         DATETIME,                       -- monday time 19
  tz_times_1_0          DATETIME,                       -- tuesday time 1
  tz_times_1_1          DATETIME,                       -- tuesday time 2
  tz_times_1_2          DATETIME,                       -- tuesday time 3
  tz_times_1_3          DATETIME,                       -- tuesday time 4
  tz_times_1_4          DATETIME,                       -- tuesday time 5
  tz_times_1_5          DATETIME,                       -- tuesday time 6
  tz_times_1_6          DATETIME,                       -- tuesday time 7
  tz_times_1_7          DATETIME,                       -- tuesday time 8
  tz_times_1_8          DATETIME,                       -- tuesday time 9
  tz_times_1_9          DATETIME,                       -- tuesday time 10
  tz_times_1_10         DATETIME,                       -- tuesday time 11
  tz_times_1_11         DATETIME,                       -- tuesday time 12
  tz_times_1_12         DATETIME,                       -- tuesday time 13
  tz_times_1_13         DATETIME,                       -- tuesday time 14
  tz_times_1_14         DATETIME,                       -- tuesday time 15
  tz_times_1_15         DATETIME,                       -- tuesday time 16
  tz_times_1_16         DATETIME,                       -- tuesday time 17
  tz_times_1_17         DATETIME,                       -- tuesday time 18
  tz_times_1_18         DATETIME,                       -- tuesday time 19
  tz_times_2_0          DATETIME,                       -- Wednesday time 1
  tz_times_2_1          DATETIME,                       -- Wednesday time 2
  tz_times_2_2          DATETIME,                       -- Wednesday time 3
  tz_times_2_3          DATETIME,                       -- Wednesday time 4
  tz_times_2_4          DATETIME,                       -- Wednesday time 5
  tz_times_2_5          DATETIME,                       -- Wednesday time 6
  tz_times_2_6          DATETIME,                       -- Wednesday time 7
  tz_times_2_7          DATETIME,                       -- Wednesday time 8
  tz_times_2_8          DATETIME,                       -- Wednesday time 9
```

```
tz_times_2_9        DATETIME,   -- Wednesday time 10
tz_times_2_10       DATETIME,   -- Wednesday time 11
tz_times_2_11       DATETIME,   -- Wednesday time 12
tz_times_2_12       DATETIME,   -- Wednesday time 13
tz_times_2_13       DATETIME,   -- Wednesday time 14
tz_times_2_14       DATETIME,   -- Wednesday time 15
tz_times_2_15       DATETIME,   -- Wednesday time 16
tz_times_2_16       DATETIME,   -- Wednesday time 17
tz_times_2_17       DATETIME,   -- Wednesday time 18
tz_times_2_18       DATETIME,   -- Wednesday time 19
tz_times_3_0        DATETIME,   -- Thursday time 1
tz_times_3_1        DATETIME,   -- Thursday time 2
tz_times_3_2        DATETIME,   -- Thursday time 3
tz_times_3_3        DATETIME,   -- Thursday time 4
tz_times_3_4        DATETIME,   -- Thursday time 5
tz_times_3_5        DATETIME,   -- Thursday time 6
tz_times_3_6        DATETIME,   -- Thursday time 7
tz_times_3_7        DATETIME,   -- Thursday time 8
tz_times_3_8        DATETIME,   -- Thursday time 9
tz_times_3_9        DATETIME,   -- Thursday time 10
tz_times_3_10       DATETIME,   -- Thursday time 11
tz_times_3_11       DATETIME,   -- Thursday time 12
tz_times_3_12       DATETIME,   -- Thursday time 13
tz_times_3_13       DATETIME,   -- Thursday time 14
tz_times_3_14       DATETIME,   -- Thursday time 15
tz_times_3_15       DATETIME,   -- Thursday time 16
tz_times_3_16       DATETIME,   -- Thursday time 17
tz_times_3_17       DATETIME,   -- Thursday time 18
tz_times_3_18       DATETIME,   -- Thursday time 19
tz_times_4_0        DATETIME,   -- Friday time 1
tz_times_4_1        DATETIME,   -- Friday time 2
tz_times_4_2        DATETIME,   -- Friday time 3
tz_times_4_3        DATETIME,   -- Friday time 4
tz_times_4_4        DATETIME,   -- Friday time 5
tz_times_4_5        DATETIME,   -- Friday time 6
tz_times_4_6        DATETIME,   -- Friday time 7
tz_times_4_7        DATETIME,   -- Friday time 8
tz_times_4_8        DATETIME,   -- Friday time 9
tz_times_4_9        DATETIME,   -- Friday time 10
tz_times_4_10       DATETIME,   -- Friday time 11
tz_times_4_11       DATETIME,   -- Friday time 12
tz_times_4_12       DATETIME,   -- Friday time 13
tz_times_4_13       DATETIME,   -- Friday time 14
tz_times_4_14       DATETIME,   -- Friday time 15
tz_times_4_15       DATETIME,   -- Friday time 16
tz_times_4_16       DATETIME,   -- Friday time 17
tz_times_4_17       DATETIME,   -- Friday time 18
tz_times_4_18       DATETIME,   -- Friday time 19
tz_times_5_0        DATETIME,   -- Saturday time 1
tz_times_5_1        DATETIME,   -- Saturday time 2
tz_times_5_2        DATETIME,   -- Saturday time 3
tz_times_5_3        DATETIME,   -- Saturday time 4
tz_times_5_4        DATETIME,   -- Saturday time 5
tz_times_5_5        DATETIME,   -- Saturday time 6
tz_times_5_6        DATETIME,   -- Saturday time 7
tz_times_5_7        DATETIME,   -- Saturday time 8
tz_times_5_8        DATETIME,   -- Saturday time 9
tz_times_5_9        DATETIME,   -- Saturday time 10
tz_times_5_10       DATETIME,   -- Saturday time 11
tz_times_5_11       DATETIME,   -- Saturday time 12
tz_times_5_12       DATETIME,   -- Saturday time 13
tz_times_5_13       DATETIME,   -- Saturday time 14
tz_times_5_14       DATETIME,   -- Saturday time 15
tz_times_5_15       DATETIME,   -- Saturday time 16
tz_times_5_16       DATETIME,   -- Saturday time 17
```

```
tz_times_5_17       DATETIME,       -- Saturday time 18
tz_times_5_18       DATETIME,       -- Saturday time 19
tz_times_6_0        DATETIME,       -- Sunday time 1
tz_times_6_1        DATETIME,       -- Sunday time 2
tz_times_6_2        DATETIME,       -- Sunday time 3
tz_times_6_3        DATETIME,       -- Sunday time 4
tz_times_6_4        DATETIME,       -- Sunday time 5
tz_times_6_5        DATETIME,       -- Sunday time 6
tz_times_6_6        DATETIME,       -- Sunday time 7
tz_times_6_7        DATETIME,       -- Sunday time 8
tz_times_6_8        DATETIME,       -- Sunday time 9
tz_times_6_9        DATETIME,       -- Sunday time 10
tz_times_6_10       DATETIME,       -- Sunday time 11
tz_times_6_11       DATETIME,       -- Sunday time 12
tz_times_6_12       DATETIME,       -- Sunday time 13
tz_times_6_13       DATETIME,       -- Sunday time 14
tz_times_6_14       DATETIME,       -- Sunday time 15
tz_times_6_15       DATETIME,       -- Sunday time 16
tz_times_6_16       DATETIME,       -- Sunday time 17
tz_times_6_17       DATETIME,       -- Sunday time 18
tz_times_6_18       DATETIME,       -- Sunday time 19
tz_times_7_0        DATETIME,       -- Holiday 1 time 1
tz_times_7_1        DATETIME,       -- Holiday 1 time 2
tz_times_7_2        DATETIME,       -- Holiday 1 time 3
tz_times_7_3        DATETIME,       -- Holiday 1 time 4
tz_times_7_4        DATETIME,       -- Holiday 1 time 5
tz_times_7_5        DATETIME,       -- Holiday 1 time 6
tz_times_7_6        DATETIME,       -- Holiday 1 time 7
tz_times_7_7        DATETIME,       -- Holiday 1 time 8
tz_times_7_8        DATETIME,       -- Holiday 1 time 9
tz_times_7_9        DATETIME,       -- Holiday 1 time 10
tz_times_7_10       DATETIME,       -- Holiday 1 time 11
tz_times_7_11       DATETIME,       -- Holiday 1 time 12
tz_times_7_12       DATETIME,       -- Holiday 1 time 13
tz_times_7_13       DATETIME,       -- Holiday 1 time 14
tz_times_7_14       DATETIME,       -- Holiday 1 time 15
tz_times_7_15       DATETIME,       -- Holiday 1 time 16
tz_times_7_16       DATETIME,       -- Holiday 1 time 17
tz_times_7_17       DATETIME,       -- Holiday 1 time 18
tz_times_7_18       DATETIME,       -- Holiday 1 time 19
tz_times_8_0        DATETIME,       -- Holiday 2 time 1
tz_times_8_1        DATETIME,       -- Holiday 2 time 2
tz_times_8_2        DATETIME,       -- Holiday 2 time 3
tz_times_8_3        DATETIME,       -- Holiday 2 time 4
tz_times_8_4        DATETIME,       -- Holiday 2 time 5
tz_times_8_5        DATETIME,       -- Holiday 2 time 6
tz_times_8_6        DATETIME,       -- Holiday 2 time 7
tz_times_8_7        DATETIME,       -- Holiday 2 time 8
tz_times_8_8        DATETIME,       -- Holiday 2 time 9
tz_times_8_9        DATETIME,       -- Holiday 2 time 10
tz_times_8_10       DATETIME,       -- Holiday 2 time 11
tz_times_8_11       DATETIME,       -- Holiday 2 time 12
tz_times_8_12       DATETIME,       -- Holiday 2 time 13
tz_times_8_13       DATETIME,       -- Holiday 2 time 14
tz_times_8_14       DATETIME,       -- Holiday 2 time 15
tz_times_8_15       DATETIME,       -- Holiday 2 time 16
tz_times_8_16       DATETIME,       -- Holiday 2 time 17
tz_times_8_17       DATETIME,       -- Holiday 2 time 18
tz_times_8_18       DATETIME,       -- Holiday 2 time 19
tz_times_9_0        DATETIME,       -- Holiday 3 time 1
tz_times_9_1        DATETIME,       -- Holiday 3 time 2
tz_times_9_2        DATETIME,       -- Holiday 3 time 3
tz_times_9_3        DATETIME,       -- Holiday 3 time 4
tz_times_9_4        DATETIME,       -- Holiday 3 time 5
tz_times_9_5        DATETIME,       -- Holiday 3 time 6
```

```
  tz_times_9_6          DATETIME,                                       -- Holiday 3 time 7
  tz_times_9_7          DATETIME,                                       -- Holiday 3 time 8
  tz_times_9_8          DATETIME,                                       -- Holiday 3 time 9
  tz_times_9_9          DATETIME,                                       -- Holiday 3 time 10
  tz_times_9_10         DATETIME,                                       -- Holiday 3 time 11
  tz_times_9_11         DATETIME,                                       -- Holiday 3 time 12
  tz_times_9_12         DATETIME,                                       -- Holiday 3 time 13
  tz_times_9_13         DATETIME,                                       -- Holiday 3 time 14
  tz_times_9_14         DATETIME,                                       -- Holiday 3 time 15
  tz_times_9_15         DATETIME,                                       -- Holiday 3 time 16
  tz_times_9_16         DATETIME,                                       -- Holiday 3 time 17
  tz_times_9_17         DATETIME,                                       -- Holiday 3 time 18
  tz_times_9_18         DATETIME,                                       -- Holiday 3 time 19
  site                  CHAR(32) NOT NULL,                              -- Site Name
  tz_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
  tz_checksum           INTEGER NOT NULL,                               -- Record checksum
  PRIMARY KEY           ( tz_partition, tz_id )
);
  CREATE INDEX              tz_public ON timezone ( tz_public );
  CREATE UNIQUE INDEX       tz_site_name ON timezone (site, tz_name);
```

CREATE TABLE timezone_save

```
(
  tz_partition          INTEGER,                                        -- partition ID to which this timezone belongs
  tz_public             SMALLINT,                                       -- 1 = public, 0 = private
  tz_id                 INTEGER NOT NULL,                               -- unique ID for this timezone
  tz_name               CHAR(32) NOT NULL,                              -- timezone name
  tz_init_stat_0        SMALLINT,                                       -- 1 = monday initial status active
  tz_init_stat_1        SMALLINT,                                       -- 1 = tuesday initial status active
  tz_init_stat_2        SMALLINT,                                       -- 1 = wednesday initial status active
  tz_init_stat_3        SMALLINT,                                       -- 1 = thursday initial status active
  tz_init_stat_4        SMALLINT,                                       -- 1 = friday initial status active
  tz_init_stat_5        SMALLINT,                                       -- 1 = saturday initial status active
  tz_init_stat_6        SMALLINT,                                       -- 1 = sunday initial status active
  tz_init_stat_7        SMALLINT,                                       -- 1 = holiday 1 initial status active
  tz_init_stat_8        SMALLINT,                                       -- 1 = holiday 2 initial status active
  tz_init_stat_9        SMALLINT,                                       -- 1 = holiday 3 initial status active
  tz_times_0_0          DATETIME,                                       -- monday time 1
  tz_times_0_1          DATETIME,                                       -- monday time 2
  tz_times_0_2          DATETIME,                                       -- monday time 3
  tz_times_0_3          DATETIME,                                       -- monday time 4
  tz_times_0_4          DATETIME,                                       -- monday time 5
  tz_times_0_5          DATETIME,                                       -- monday time 6
  tz_times_0_6          DATETIME,                                       -- monday time 7
  tz_times_0_7          DATETIME,                                       -- monday time 8
  tz_times_0_8          DATETIME,                                       -- monday time 9
  tz_times_0_9          DATETIME,                                       -- monday time 10
  tz_times_0_10         DATETIME,                                       -- monday time 11
  tz_times_0_11         DATETIME,                                       -- monday time 12
  tz_times_0_12         DATETIME,                                       -- monday time 13
  tz_times_0_13         DATETIME,                                       -- monday time 14
  tz_times_0_14         DATETIME,                                       -- monday time 15
  tz_times_0_15         DATETIME,                                       -- monday time 16
  tz_times_0_16         DATETIME,                                       -- monday time 17
  tz_times_0_17         DATETIME,                                       -- monday time 18
  tz_times_0_18         DATETIME,                                       -- monday time 19
  tz_times_1_0          DATETIME,                                       -- tuesday time 1
  tz_times_1_1          DATETIME,                                       -- tuesday time 2
  tz_times_1_2          DATETIME,                                       -- tuesday time 3
  tz_times_1_3          DATETIME,                                       -- tuesday time 4
  tz_times_1_4          DATETIME,                                       -- tuesday time 5
  tz_times_1_5          DATETIME,                                       -- tuesday time 6
  tz_times_1_6          DATETIME,                                       -- tuesday time 7
```

```
tz_times_1_7        DATETIME,    -- tuesday time 8
tz_times_1_8        DATETIME,    -- tuesday time 9
tz_times_1_9        DATETIME,    -- tuesday time 10
tz_times_1_10       DATETIME,    -- tuesday time 11
tz_times_1_11       DATETIME,    -- tuesday time 12
tz_times_1_12       DATETIME,    -- tuesday time 13
tz_times_1_13       DATETIME,    -- tuesday time 14
tz_times_1_14       DATETIME,    -- tuesday time 15
tz_times_1_15       DATETIME,    -- tuesday time 16
tz_times_1_16       DATETIME,    -- tuesday time 17
tz_times_1_17       DATETIME,    -- tuesday time 18
tz_times_1_18       DATETIME,    -- tuesday time 19
tz_times_2_0        DATETIME,    -- Wednesday time 1
tz_times_2_1        DATETIME,    -- Wednesday time 2
tz_times_2_2        DATETIME,    -- Wednesday time 3
tz_times_2_3        DATETIME,    -- Wednesday time 4
tz_times_2_4        DATETIME,    -- Wednesday time 5
tz_times_2_5        DATETIME,    -- Wednesday time 6
tz_times_2_6        DATETIME,    -- Wednesday time 7
tz_times_2_7        DATETIME,    -- Wednesday time 8
tz_times_2_8        DATETIME,    -- Wednesday time 9
tz_times_2_9        DATETIME,    -- Wednesday time 10
tz_times_2_10       DATETIME,    -- Wednesday time 11
tz_times_2_11       DATETIME,    -- Wednesday time 12
tz_times_2_12       DATETIME,    -- Wednesday time 13
tz_times_2_13       DATETIME,    -- Wednesday time 14
tz_times_2_14       DATETIME,    -- Wednesday time 15
tz_times_2_15       DATETIME,    -- Wednesday time 16
tz_times_2_16       DATETIME,    -- Wednesday time 17
tz_times_2_17       DATETIME,    -- Wednesday time 18
tz_times_2_18       DATETIME,    -- Wednesday time 19
tz_times_3_0        DATETIME,    -- Thursday time 1
tz_times_3_1        DATETIME,    -- Thursday time 2
tz_times_3_2        DATETIME,    -- Thursday time 3
tz_times_3_3        DATETIME,    -- Thursday time 4
tz_times_3_4        DATETIME,    -- Thursday time 5
tz_times_3_5        DATETIME,    -- Thursday time 6
tz_times_3_6        DATETIME,    -- Thursday time 7
tz_times_3_7        DATETIME,    -- Thursday time 8
tz_times_3_8        DATETIME,    -- Thursday time 9
tz_times_3_9        DATETIME,    -- Thursday time 10
tz_times_3_10       DATETIME,    -- Thursday time 11
tz_times_3_11       DATETIME,    -- Thursday time 12
tz_times_3_12       DATETIME,    -- Thursday time 13
tz_times_3_13       DATETIME,    -- Thursday time 14
tz_times_3_14       DATETIME,    -- Thursday time 15
tz_times_3_15       DATETIME,    -- Thursday time 16
tz_times_3_16       DATETIME,    -- Thursday time 17
tz_times_3_17       DATETIME,    -- Thursday time 18
tz_times_3_18       DATETIME,    -- Thursday time 19
tz_times_4_0        DATETIME,    -- Friday time 1
tz_times_4_1        DATETIME,    -- Friday time 2
tz_times_4_2        DATETIME,    -- Friday time 3
tz_times_4_3        DATETIME,    -- Friday time 4
tz_times_4_4        DATETIME,    -- Friday time 5
tz_times_4_5        DATETIME,    -- Friday time 6
tz_times_4_6        DATETIME,    -- Friday time 7
tz_times_4_7        DATETIME,    -- Friday time 8
tz_times_4_8        DATETIME,    -- Friday time 9
tz_times_4_9        DATETIME,    -- Friday time 10
tz_times_4_10       DATETIME,    -- Friday time 11
tz_times_4_11       DATETIME,    -- Friday time 12
tz_times_4_12       DATETIME,    -- Friday time 13
tz_times_4_13       DATETIME,    -- Friday time 14
tz_times_4_14       DATETIME,    -- Friday time 15
```

```
tz_times_4_15       DATETIME,       -- Friday time 16
tz_times_4_16       DATETIME,       -- Friday time 17
tz_times_4_17       DATETIME,       -- Friday time 18
tz_times_4_18       DATETIME,       -- Friday time 19
tz_times_5_0        DATETIME,       -- Saturday time 1
tz_times_5_1        DATETIME,       -- Saturday time 2
tz_times_5_2        DATETIME,       -- Saturday time 3
tz_times_5_3        DATETIME,       -- Saturday time 4
tz_times_5_4        DATETIME,       -- Saturday time 5
tz_times_5_5        DATETIME,       -- Saturday time 6
tz_times_5_6        DATETIME,       -- Saturday time 7
tz_times_5_7        DATETIME,       -- Saturday time 8
tz_times_5_8        DATETIME,       -- Saturday time 9
tz_times_5_9        DATETIME,       -- Saturday time 10
tz_times_5_10       DATETIME,       -- Saturday time 11
tz_times_5_11       DATETIME,       -- Saturday time 12
tz_times_5_12       DATETIME,       -- Saturday time 13
tz_times_5_13       DATETIME,       -- Saturday time 14
tz_times_5_14       DATETIME,       -- Saturday time 15
tz_times_5_15       DATETIME,       -- Saturday time 16
tz_times_5_16       DATETIME,       -- Saturday time 17
tz_times_5_17       DATETIME,       -- Saturday time 18
tz_times_5_18       DATETIME,       -- Saturday time 19
tz_times_6_0        DATETIME,       -- Sunday time 1
tz_times_6_1        DATETIME,       -- Sunday time 2
tz_times_6_2        DATETIME,       -- Sunday time 3
tz_times_6_3        DATETIME,       -- Sunday time 4
tz_times_6_4        DATETIME,       -- Sunday time 5
tz_times_6_5        DATETIME,       -- Sunday time 6
tz_times_6_6        DATETIME,       -- Sunday time 7
tz_times_6_7        DATETIME,       -- Sunday time 8
tz_times_6_8        DATETIME,       -- Sunday time 9
tz_times_6_9        DATETIME,       -- Sunday time 10
tz_times_6_10       DATETIME,       -- Sunday time 11
tz_times_6_11       DATETIME,       -- Sunday time 12
tz_times_6_12       DATETIME,       -- Sunday time 13
tz_times_6_13       DATETIME,       -- Sunday time 14
tz_times_6_14       DATETIME,       -- Sunday time 15
tz_times_6_15       DATETIME,       -- Sunday time 16
tz_times_6_16       DATETIME,       -- Sunday time 17
tz_times_6_17       DATETIME,       -- Sunday time 18
tz_times_6_18       DATETIME,       -- Sunday time 19
tz_times_7_0        DATETIME,       -- Holiday 1 time 1
tz_times_7_1        DATETIME,       -- Holiday 1 time 2
tz_times_7_2        DATETIME,       -- Holiday 1 time 3
tz_times_7_3        DATETIME,       -- Holiday 1 time 4
tz_times_7_4        DATETIME,       -- Holiday 1 time 5
tz_times_7_5        DATETIME,       -- Holiday 1 time 6
tz_times_7_6        DATETIME,       -- Holiday 1 time 7
tz_times_7_7        DATETIME,       -- Holiday 1 time 8
tz_times_7_8        DATETIME,       -- Holiday 1 time 9
tz_times_7_9        DATETIME,       -- Holiday 1 time 10
tz_times_7_10       DATETIME,       -- Holiday 1 time 11
tz_times_7_11       DATETIME,       -- Holiday 1 time 12
tz_times_7_12       DATETIME,       -- Holiday 1 time 13
tz_times_7_13       DATETIME,       -- Holiday 1 time 14
tz_times_7_14       DATETIME,       -- Holiday 1 time 15
tz_times_7_15       DATETIME,       -- Holiday 1 time 16
tz_times_7_16       DATETIME,       -- Holiday 1 time 17
tz_times_7_17       DATETIME,       -- Holiday 1 time 18
tz_times_7_18       DATETIME,       -- Holiday 1 time 19
tz_times_8_0        DATETIME,       -- Holiday 2 time 1
tz_times_8_1        DATETIME,       -- Holiday 2 time 2
tz_times_8_2        DATETIME,       -- Holiday 2 time 3
tz_times_8_3        DATETIME,       -- Holiday 2 time 4
```

```
  tz_times_8_4          DATETIME,                                   -- Holiday 2 time 5
  tz_times_8_5          DATETIME,                                   -- Holiday 2 time 6
  tz_times_8_6          DATETIME,                                   -- Holiday 2 time 7
  tz_times_8_7          DATETIME,                                   -- Holiday 2 time 8
  tz_times_8_8          DATETIME,                                   -- Holiday 2 time 9
  tz_times_8_9          DATETIME,                                   -- Holiday 2 time 10
  tz_times_8_10         DATETIME,                                   -- Holiday 2 time 11
  tz_times_8_11         DATETIME,                                   -- Holiday 2 time 12
  tz_times_8_12         DATETIME,                                   -- Holiday 2 time 13
  tz_times_8_13         DATETIME,                                   -- Holiday 2 time 14
  tz_times_8_14         DATETIME,                                   -- Holiday 2 time 15
  tz_times_8_15         DATETIME,                                   -- Holiday 2 time 16
  tz_times_8_16         DATETIME,                                   -- Holiday 2 time 17
  tz_times_8_17         DATETIME,                                   -- Holiday 2 time 18
  tz_times_8_18         DATETIME,                                   -- Holiday 2 time 19
  tz_times_9_0          DATETIME,                                   -- Holiday 3 time 1
  tz_times_9_1          DATETIME,                                   -- Holiday 3 time 2
  tz_times_9_2          DATETIME,                                   -- Holiday 3 time 3
  tz_times_9_3          DATETIME,                                   -- Holiday 3 time 4
  tz_times_9_4          DATETIME,                                   -- Holiday 3 time 5
  tz_times_9_5          DATETIME,                                   -- Holiday 3 time 6
  tz_times_9_6          DATETIME,                                   -- Holiday 3 time 7
  tz_times_9_7          DATETIME,                                   -- Holiday 3 time 8
  tz_times_9_8          DATETIME,                                   -- Holiday 3 time 9
  tz_times_9_9          DATETIME,                                   -- Holiday 3 time 10
  tz_times_9_10         DATETIME,                                   -- Holiday 3 time 11
  tz_times_9_11         DATETIME,                                   -- Holiday 3 time 12
  tz_times_9_12         DATETIME,                                   -- Holiday 3 time 13
  tz_times_9_13         DATETIME,                                   -- Holiday 3 time 14
  tz_times_9_14         DATETIME,                                   -- Holiday 3 time 15
  tz_times_9_15         DATETIME,                                   -- Holiday 3 time 16
  tz_times_9_16         DATETIME,                                   -- Holiday 3 time 17
  tz_times_9_17         DATETIME,                                   -- Holiday 3 time 18
  tz_times_9_18         DATETIME,                                   -- Holiday 3 time 19
  site                  CHAR(32) NOT NULL,                          -- Site Name
  tz_guid               uniqueidentifier NOT NULL,                  -- Global unique record identifier
  tz_checksum           INTEGER NOT NULL,                           -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                           -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid()
                          UNIQUE NOT NULL,                          -- Global unique record identifier
  PRIMARY KEY           ( tz_partition, tz_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE tour_configuration

```
(
  tc_partition          INTEGER REFERENCES partition(part_number)
                          NOT FOR REPLICATION,                      -- partition ID to which this tour belongs
  tc_public             SMALLINT,                                   -- 1 = tour is public, 0 = tour is private
  tc_tour_id            INTEGER IDENTITY NOT FOR REPLICATION UNIQUE, -- unique database ID
  tc_tour_name          CHAR(32) NOT NULL,                          -- user entered name
  tc_description        CHAR(64),
  tc_type               SMALLINT NOT NULL,                          -- tour type. 0 - Manual, 1-AutoForward,
                                                                      2-AutoReverse, 3-Random, 4-ManualForward,
                                                                      5-ManualReverse
  tc_priority           SMALLINT NOT NULL,                          -- tour priority
  tc_time_zone          INTEGER,                                    -- assigned timezone
  tc_time_zone_name     CHAR(32),                                   -- timezone name
  tc_start_time         DATETIME,                                   -- defined start time
  tc_abort_run_time     INTEGER,                                    -- abort time or run time for Random tours
  tc_alarm_priority     INTEGER,                                    -- alarm priority
```

```
    tc_tour_guard1              INTEGER,                                    -- selected guard
    tc_tour_guard1_name         CHAR(52),                                   -- guard name
    tc_guard_badge_id1          CHAR(32),                                   -- badge id
    tc_tour_guard2              INTEGER,
    tc_tour_guard2_name         CHAR(52),
    tc_guard_badge_id2          CHAR(32),
    tc_tour_guard3              INTEGER,
    tc_tour_guard3_name         CHAR(52),
    tc_guard_badge_id3          CHAR(32),
    tc_alarm_late               SMALLINT,                                   -- set alarm if tour is late
    tc_auto_duress_alrarm       SMALLINT,                                   -- generate duress alarm if three check-ins at the
                                                                               same station in one minute
    tc_log_oper_action          SMALLINT,                                   -- log operator actions
    tc_log_tour_oper            SMALLINT,                                   -- log tour operations
    tc_any_guard                SMALLINT,                                   -- whether any guard can run this tour
    tc_manual_reset             SMALLINT,                                   -- whether manual reset
    tc_alarm_skip               SMALLINT,                                   -- set alarm if station is skipped
    tc_popup_on_late_alarm      SMALLINT NOT NULL,                          -- 0 = do not pop up, 1 = pop up on alarm,
                                                                               2 = pop up on secure, 3 = pop up on both
    tc_popup_on_out_of_seq_alarm SMALLINT NOT NULL,                         -- Same for the late alarm
    tc_popup_on_duress_alarm    SMALLINT NOT NULL,                          -- Same for the late alarm
    tc_late_alarm_text_set_id   INTEGER NULL REFERENCES alrminst(ai_id)
                                   NOT FOR REPLICATION,                     -- late alarm set instruction text ID if not null
    tc_late_alarm_text_sec_id   INTEGER NULL REFERENCES alrminst(ai_id)
                                   NOT FOR REPLICATION,                     -- late alarm secure instruction text
    tc_out_of_seq_alarm_text_set_id INTEGER NULL REFERENCES alrminst(ai_id)
                                   NOT FOR REPLICATION,                     -- Same for the late alarm
    tc_out_of_seq_alarm_text_sec_id INTEGER NULL REFERENCES alrminst(ai_id)
                                   NOT FOR REPLICATION,                     -- Etc.
    tc_duress_alarm_text_set_id INTEGER NULL REFERENCES alrminst(ai_id)
                                   NOT FOR REPLICATION,                     -- Same for the late alarm
    tc_duress_alarm_text_sec_id INTEGER NULL REFERENCES alrminst(ai_id)
                                   NOT FOR REPLICATION,                     -- Etc
    tc_grant_only               SMALLINT,                                   -- take grant messages only or not
    site                        CHAR(32) NOT NULL,                          -- Site Name
    tc_guid                     uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
    tc_checksum                 INTEGER NOT NULL,                           -- Record checksum
    PRIMARY KEY                 ( tc_partition, tc_tour_id )
);
    CREATE UNIQUE INDEX         tc_site_name ON tour_configuration (site, tc_tour_name);
```

CREATE TABLE tour_configuration_save

```
(
    tc_partition                INTEGER,                                    -- partition ID to which this tour belongs
    tc_public                   SMALLINT,                                   -- 1 = tour is public, 0 = tour is private
    tc_tour_id                  INTEGER NOT NULL,                           -- unique database ID
    tc_tour_name                CHAR(32) NOT NULL,                          -- user entered name
    tc_description              CHAR(64),
    tc_type                     SMALLINT NOT NULL,                          -- tour type. 0 - Manual, 1-AutoForward, 2-AutoReverse,
                                                                               3-Random, 4-ManualForward, 5-ManualReverse
    tc_priority                 SMALLINT NOT NULL,                          -- tour priority
    tc_time_zone                INTEGER,                                    -- assigned timezone
    tc_time_zone_name           CHAR(32),                                   -- timezone name
    tc_start_time               DATETIME,                                   -- defined start time
    tc_abort_run_time           INTEGER,                                    -- abort time or run time for Random tours
    tc_alarm_priority           INTEGER,                                    -- alarm priority
    tc_tour_guard1              INTEGER,                                    -- selected guard
    tc_tour_guard1_name         CHAR(52),                                   -- guard name
    tc_guard_badge_id1          CHAR(32),                                   -- badge id
    tc_tour_guard2              INTEGER,
    tc_tour_guard2_name         CHAR(52),
    tc_guard_badge_id2          CHAR(32),
    tc_tour_guard3              INTEGER,
```

```
  tc_tour_guard3_name           CHAR(52),
  tc_guard_badge_id3            CHAR(32),
  tc_alarm_late                 SMALLINT,                    -- set alarm if tour is late
  tc_auto_duress_alrarm         SMALLINT,                    -- generate duress alarm if three check-ins at the same station in
                                                                one minute
  tc_log_oper_action            SMALLINT,                    -- log operator actions
  tc_log_tour_oper              SMALLINT,                    -- log tour operations
  tc_any_guard                  SMALLINT,                    -- whether any guard can run this tour
  tc_manual_reset               SMALLINT,                    -- whether manual reset
  tc_alarm_skip                 SMALLINT,                    -- set alarm if station is skipped
  tc_popup_on_late_alarm        SMALLINT NOT NULL,           -- 0 = do not pop up, 1 = pop up on alarm,
                                                                2 = pop up on secure, 3 = pop up on both
  tc_popup_on_out_of_seq_alarm  SMALLINT NOT NULL,           -- Same for the late alarm
  tc_popup_on_duress_alarm      SMALLINT NOT NULL,           -- Same for the late alarm
  tc_late_alarm_text_set_id     INTEGER NULL,                -- late alarm set instruction text ID if not null
  tc_late_alarm_text_sec_id     INTEGER NULL,                -- late alarm secure instruction text
  tc_out_of_seq_alarm_text_set_id INTEGER NULL,              -- Same for the late alarm
  tc_out_of_seq_alarm_text_sec_id INTEGER NULL,              -- Etc.
  tc_duress_alarm_text_set_id   INTEGER NULL,                -- Same for the late alarm
  tc_duress_alarm_text_sec_id   INTEGER NULL,                -- Etc
  tc_grant_only                 SMALLINT,                    -- take grant messages only or not
  site                          CHAR(32) NOT NULL,           -- Site Name
  tc_guid                       uniqueidentifier NOT NULL,   -- Global unique record identifier
  tc_checksum                   INTEGER NOT NULL,            -- Record checksum
  upd_guid                      uniqueidentifier NOT NULL,   -- Guid for update operation
  upd_mode                      SMALLINT NOT NULL,           -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                 DATETIME NOT NULL,           -- Date / Time of the update
  upd_checksum                  INTEGER NOT NULL,            -- Checksum
  upd_rowguid                   uniqueidentifier ROWGUIDCOL DEFAULT
                                  newid() UNIQUE NOT NULL,   -- Global unique record identifier
  PRIMARY KEY                   ( tc_partition, tc_tour_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE tour_note

```
(
  tn_id             INTEGER IDENTITY NOT FOR REPLICATION PRIMARY KEY,
  tn_tour_name      CHAR(32) NULL,
  tn_text           CHAR(255) NULL,
  tn_username       CHAR(33) NULL,
  tn_timestamp      DATETIME NULL,
  site              CHAR(32) NOT NULL,                       -- Site Name
  tn_guid           uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  tn_checksum       INTEGER NOT NULL,                        -- Record checksum
);
  CREATE INDEX      tn_tour_name ON tour_note (tn_tour_name);
  CREATE INDEX      tn_timestamp ON tour_note (tn_timestamp);
```

CREATE TABLE tour_station_configuration

```
(
  tsc_tour_id       INTEGER REFERENCES tour_configuration (tc_tour_id)
                      NOT FOR REPLICATION,                   -- tour id this station belongs to
  tsc_sequence_num  INTEGER NOT NULL,                        -- sequence number
  tsc_station_name  CHAR(32) NOT NULL,                       -- user entered name
  tsc_description   CHAR(64),
  tsc_type          SMALLINT NOT NULL,                       -- station type. 0-Input, 1-Reader
  tsc_device_id     INTEGER,                                 -- the input or terminal id this station is assigned
  tsc_device_name   CHAR(32),
  tsc_server_type   SMALLINT,                                -- server type
  tsc_server_id     CHAR(32),                                -- server id
  tsc_next_min_timer INTEGER,                                -- the min time to reach this station forwardly
  tsc_next_max_timer INTEGER,                                -- the max time to reach this station forwardly
  tsc_prev_min_timer INTEGER,                                -- the min time to reach this station backwardly
```

```
    tsc_prev_max_timer          INTEGER,                                        -- the max time to reach this station backwardly
    tsc_input_object_id         INTEGER,                                        -- the suppress input id
    tsc_input_object_name       CHAR(32),                                       -- the suppress input name
    tsc_input_object_type       SMALLINT,                                       -- the suppress type. 0-input point, 1-input group
    tsc_input_object_attr       SMALLINT,                                       -- the suppress input attribute
    tsc_output_object_id1       INTEGER,                                        -- the shunt output id
    tsc_output_object_name1     CHAR(32),                                       -- the shunt output name
    tsc_output_object_type1     SMALLINT,                                       -- the shunt type. 0-output point, 1-output group
    tsc_output_object_attr1     SMALLINT,                                       -- the shunt output attribute
    tsc_output_object_value1    INTEGER,                                        -- the shunt output value
    tsc_output_object_id2       INTEGER,
    tsc_output_object_name2     CHAR(32),
    tsc_output_object_type2     SMALLINT,
    tsc_output_object_attr2     SMALLINT,
    tsc_output_object_value2    INTEGER,
    tsc_output_object_id3       INTEGER,
    tsc_output_object_name3     CHAR(32),
    tsc_output_object_type3     SMALLINT,
    tsc_output_object_attr3     SMALLINT,
    tsc_output_object_value3    INTEGER,
    tsc_output_object_id4       INTEGER,
    tsc_output_object_name4     CHAR(32),
    tsc_output_object_type4     SMALLINT,
    tsc_output_object_attr4     SMALLINT,
    tsc_output_object_value4    INTEGER,
    site                        CHAR(32) NOT NULL,                              -- Site Name
    tsc_guid                    uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
    tsc_checksum                INTEGER NOT NULL,                               -- Record checksum
    PRIMARY KEY                 (tsc_tour_id, tsc_sequence_num)
);
    CREATE INDEX                tsc_tour_id ON tour_station_configuration (tsc_tour_id);
    CREATE INDEX                tsc_station_name ON tour_station_configuration (tsc_station_name);
```

CREATE TABLE tour_station_configuration_save

```
(
    tsc_tour_id                 INTEGER,                                        -- tour id this station belongs to
    tsc_sequence_num            INTEGER NOT NULL,                               -- sequence number
    tsc_station_name            CHAR(32) NOT NULL,                              -- user entered name
    tsc_description             CHAR(64),
    tsc_type                    SMALLINT NOT NULL,                              -- station type. 0-Input, 1-Reader
    tsc_device_id               INTEGER,                                        -- the input or terminal id this station is assigned
    tsc_device_name             CHAR(32),
    tsc_server_type             SMALLINT,                                       -- server type
    tsc_server_id               CHAR(32),                                       -- server id
    tsc_next_min_timer          INTEGER,                                        -- the min time to reach this station forwardly
    tsc_next_max_timer          INTEGER,                                        -- the max time to reach this station forwardly
    tsc_prev_min_timer          INTEGER,                                        -- the min time to reach this station backwardly
    tsc_prev_max_timer          INTEGER,                                        -- the max time to reach this station backwardly
    tsc_input_object_id         INTEGER,                                        -- the suppress input id
    tsc_input_object_name       CHAR(32),                                       -- the suppress input name
    tsc_input_object_type       SMALLINT,                                       -- the suppress type. 0-input point, 1-input group
    tsc_input_object_attr       SMALLINT,                                       -- the suppress input attribute
    tsc_output_object_id1       INTEGER,                                        -- the shunt output id
    tsc_output_object_name1     CHAR(32),                                       -- the shunt output name
    tsc_output_object_type1     SMALLINT,                                       -- the shunt type. 0-output point, 1-output group
    tsc_output_object_attr1     SMALLINT,                                       -- the shunt output attribute
    tsc_output_object_value1    INTEGER,                                        -- the shunt output value
    tsc_output_object_id2       INTEGER,
    tsc_output_object_name2     CHAR(32),
    tsc_output_object_type2     SMALLINT,
    tsc_output_object_attr2     SMALLINT,
    tsc_output_object_value2    INTEGER,
    tsc_output_object_id3       INTEGER,
```

```
  tsc_output_object_name3     CHAR(32),
  tsc_output_object_type3     SMALLINT,
  tsc_output_object_attr3     SMALLINT,
  tsc_output_object_value3    INTEGER,
  tsc_output_object_id4       INTEGER,
  tsc_output_object_name4     CHAR(32),
  tsc_output_object_type4     SMALLINT,
  tsc_output_object_attr4     SMALLINT,
  tsc_output_object_value4    INTEGER,
  site                        CHAR(32) NOT NULL,                          -- Site Name
  tsc_guid                    uniqueidentifier NOT NULL,                  -- Global unique record identifier
  tsc_checksum                INTEGER NOT NULL,                           -- Record checksum
  upd_guid                    uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode                    SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp               DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum                INTEGER NOT NULL,                           -- Checksum
  upd_rowguid                 uniqueidentifier ROWGUIDCOL DEFAULT newid()
                                 UNIQUE NOT NULL,                         -- Global unique record identifier
  PRIMARY KEY                 (tsc_tour_id, tsc_sequence_num, upd_mode desc, upd_guid)
);
```

CREATE TABLE tour_status

```
(
  ts_tour_id                  INTEGER REFERENCES tour_configuration (tc_tour_id)
                                 NOT FOR REPLICATION,
  ts_tour_status              SMALLINT NOT NULL,                          -- tour status
  ts_start_time               DATETIME,                                   -- start time
  ts_direction                SMALLINT NOT NULL,                          -- direction of the running tour, 0 - forward, 1 - reverse,
                                                                             2 - random
  ts_last_station             INTEGER,                                    -- last check-in station
  ts_last_station_time        DATETIME,                                   -- last check-in time
  ts_remain_time              INTEGER,                                    -- the remaining time to reach the next station
  ts_guard_name               CHAR(52),                                   -- guard name
  ts_guard_badge_id           CHAR(32),                                   -- guard badge id
  ts_starting_station         INTEGER,                                    -- start station
  ts_last_status              SMALLINT,                                   -- the previous tour status
  ts_alarm_station            INTEGER,                                    -- the alarmed station
  ts_operator                 CHAR(32),                                   -- the operator who makes the action
  site                        CHAR(32) NOT NULL,                          -- Site Name
  ts_guid                     uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  ts_checksum                 INTEGER NOT NULL,                           -- Record checksum
  PRIMARY KEY                 (ts_tour_id)
);
```

CREATE TABLE trigger_event_p900

```
(
  tep_id                      INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,  --Unique trigger event ID
  tep_name                    varchar(32),                                  -- the trigger event name.
  tep_number                  SMALLINT,                                     -- trigger event number
  tep_status                  SMALLINT NOT NULL,                            -- trigger event status
  tep_selected                SMALLINT,                                     -- trigger event selected or not
  tep_source_type             INTEGER,                                      -- source event type
  tep_source_id               INTEGER,                                      -- source event terminal ID
  tep_source_para1            INTEGER,                                      -- source event site code
  tep_source_para2            INTEGER,                                      -- source event access level
  tep_source_para3            INTEGER,                                      -- source event card number
  tep_source_flag1            INTEGER NULL REFERENCES flag_p900(flp_id)
                                 NOT FOR REPLICATION,                       -- source event flag #1
  tep_source_flag1_state      SMALLINT,                                     -- state of source event flag #1
  tep_source_flag2            INTEGER NULL REFERENCES flag_p900(flp_id)
                                 NOT FOR REPLICATION,                       -- source event flag #2
  tep_source_flag2_state      SMALLINT,                                     -- state of source event flag #2
```

```
    tep_source_counter              INTEGER NULL REFERENCES counter_p900(cp_id)
                                      NOT FOR REPLICATION,                    -- source counter
    tep_source_counter_state        SMALLINT,                                 -- counter trigger condition
    tep_source_counter_value        INTEGER,                                  -- counter current value
    tep_time_zone                   INTEGER NULL REFERENCES timezone(tz_id)
                                      NOT FOR REPLICATION,                    -- time zone
    tep_time_zone_state             SMALLINT,                                 -- time zone enable/disable
    tep_input_action                INTEGER,                                  -- input action
    tep_input_id                    INTEGER NULL REFERENCES input(ip_point_id)
                                      NOT FOR REPLICATION,                    -- input address
    tep_input_group_id              INTEGER NULL REFERENCES inputgrp(ip_group_id)
                                      NOT FOR REPLICATION,                    -- input number
    tep_output_action               INTEGER,                                  -- output action
    tep_output_id                   INTEGER NULL REFERENCES output(op_output_id)
                                      NOT FOR REPLICATION,                    -- output address
    tep_output_group_id             INTEGER NULL REFERENCES outputgrp(op_group_id)
                                      NOT FOR REPLICATION,                    -- output number
    tep_output_flag1                INTEGER NULL REFERENCES flag_p900(flp_id)
                                      NOT FOR REPLICATION,                    -- output flag1
    tep_output_flag1_state          SMALLINT,                                 -- state of output flag1
    tep_output_flag2                INTEGER NULL REFERENCES flag_p900(flp_id)
                                      NOT FOR REPLICATION,                    -- output flag2
    tep_output_flag2_state          SMALLINT,                                 -- state of output flag2
    tep_output_counter              INTEGER NULL REFERENCES counter_p900(cp_id)
                                      NOT FOR REPLICATION,                    -- output counter
    tep_output_counter_state        SMALLINT,                                 -- state of output couneter
    tep_output_counter_value        INTEGER,                                  -- value of output couneter
    tep_input_sp_value              INTEGER,                                  -- shunt period value
    tep_output_pp_value             INTEGER,                                  -- pulse period value
    tep_report_priority             SMALLINT,                                 -- priority of report message
    tep_terminal_id_for_card_number INTEGER NULL REFERENCES terminal(tp_term_id)
                                      NOT FOR REPLICATION,                    -- terminal ID for report card number
    tep_force_reconfig              SMALLINT,                                 -- 1- force 0- no force
    tep_action1                     INTEGER NULL REFERENCES alrminst(ai_id)
                                      NOT FOR REPLICATION,                    -- automatic alarm action #1
    tep_query_string                CHAR(64),                                 -- query string enter by user
    tep_partition                   INTEGER REFERENCES partition(part_number)
                                      NOT FOR REPLICATION,                    -- ID of partition for this trigger event
    tep_public                      SMALLINT NOT NULL,                        -- open to punlic
    tep_panel_id                    INTEGER NOT NULL REFERENCES panel(p_panel_id)
                                      NOT FOR REPLICATION,                    -- Panel ID
    tep_alarm_popup                 SMALLINT,                                 -- alarm instruction pop up, 1: yes, 0: No
    site                            CHAR(32) NOT NULL,                        -- Site Name
    tep_guid                        uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
    tep_checksum                    INTEGER NOT NULL,                         -- Record checksum
    PRIMARY KEY                     (tep_partition, tep_id)
);
    CREATE INDEX                    tep_public ON trigger_event_p900(tep_public);
    CREATE UNIQUE INDEX             tep_number ON trigger_event_p900(tep_number, tep_panel_id);
    CREATE UNIQUE INDEX             tep_site_name ON trigger_event_p900 (site, tep_name);
```

CREATE TABLE trigger_event_p900_save

```
(
    tep_id                          INTEGER NOT NULL,                         --Unique trigger event ID
    tep_name                        varchar(32),                              -- the trigger event name.
    tep_number                      SMALLINT,                                 -- trigger event number
    tep_status                      SMALLINT NOT NULL,                        -- trigger event status
    tep_selected                    SMALLINT,                                 -- trigger event selected or not
    tep_source_type                 INTEGER,                                  -- source event type
    tep_source_id                   INTEGER,                                  -- source event terminal ID
    tep_source_para1                INTEGER,                                  -- source event site code
    tep_source_para2                INTEGER,                                  -- source event access level
    tep_source_para3                INTEGER,                                  -- source event card number
```

```
  tep_source_flag1                INTEGER NULL,                                -- source event flag #1
  tep_source_flag1_state          SMALLINT,                                    -- state of source event flag #1
  tep_source_flag2                INTEGER NULL,                                -- source event flag #2
  tep_source_flag2_state          SMALLINT,                                    -- state of source event flag #2
  tep_source_counter              INTEGER NULL,                                -- source counter
  tep_source_counter_state        SMALLINT,                                    -- counter trigger condition
  tep_source_counter_value        INTEGER,                                     -- counter current value
  tep_time_zone                   INTEGER NULL,                                -- time zone
  tep_time_zone_state             SMALLINT,                                    -- time zone enable/disable
  tep_input_action                INTEGER,                                     -- input action
  tep_input_id                    INTEGER NULL,                                -- input address
  tep_input_group_id              INTEGER NULL,                                -- input number
  tep_output_action               INTEGER,                                     -- output action
  tep_output_id                   INTEGER NULL,                                -- output address
  tep_output_group_id             INTEGER NULL,                                -- output number
  tep_output_flag1                INTEGER NULL,                                -- output flag1
  tep_output_flag1_state          SMALLINT,                                    -- state of output flag1
  tep_output_flag2                INTEGER NULL,                                -- output flag2
  tep_output_flag2_state          SMALLINT,                                    -- state of output flag2
  tep_output_counter              INTEGER NULL,                                -- output counter
  tep_output_counter_state        SMALLINT,                                    -- state of output couneter
  tep_output_counter_value        INTEGER,                                     -- value of output couneter
  tep_input_sp_value              INTEGER,                                     -- shunt period value
  tep_output_pp_value             INTEGER,                                     -- pulse period value
  tep_report_priority             SMALLINT,                                    -- priority of report  message
  tep_terminal_id_for_card_number INTEGER NULL,                                -- terminal ID for report card number
  tep_force_reconfig              SMALLINT,                                    -- 1- force 0- no  force
  tep_action1                     INTEGER NULL,                                -- automatic alarm action #1
  tep_query_string                CHAR(64),                                    -- query string enter by user
  tep_partition                   INTEGER,                                     -- ID of partition for this trigger event
  tep_public                      SMALLINT NOT NULL,                           -- open to punlic
  tep_panel_id                    INTEGER NOT NULL,                            -- Panel ID
  tep_alarm_popup                 SMALLINT,                                    -- alarm instruction pop up, 1: yes, 0: No
  site                            CHAR(32) NOT NULL,                           -- Site Name
  tep_guid                        uniqueidentifier NOT NULL,                   -- Global unique record identifier
  tep_checksum                    INTEGER NOT NULL,                            -- Record checksum
  upd_guid                        uniqueidentifier NOT NULL,                   -- Guid for update operation
  upd_mode                        SMALLINT NOT NULL,                           -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                   DATETIME NOT NULL,                           -- Date / Time of the update
  upd_checksum                    INTEGER NOT NULL,                            -- Checksum
  upd_rowguid                     uniqueidentifier ROWGUIDCOL DEFAULT newid()
                                    UNIQUE NOT NULL,                           -- Global unique record identifier
  PRIMARY KEY                     (tep_partition, tep_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE trigger_link_p900

```
(
  tlp_id                  INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,         -- unique ID
  tlp_name                varchar(32),                                         -- the link name.
  tlp_enable              SMALLINT NOT NULL,                                   -- enable
  tlp_source_panel        INTEGER NOT NULL REFERENCES panel(p_panel_id)
                            NOT FOR REPLICATION,                               -- source controller
  tlp_source_event        INTEGER NOT NULL REFERENCES trigger_event_p900(tep_id)
                            NOT FOR REPLICATION,                               -- source event
  tlp_destination_panel   INTEGER NOT NULL REFERENCES panel(p_panel_id)
                            NOT FOR REPLICATION,                               -- destination controller
  tlp_query_string        CHAR(64),                                            -- query string enter by user
  tlp_destination_event   INTEGER NOT NULL REFERENCES trigger_event_p900(tep_id)
                            NOT FOR REPLICATION,                               -- destination event
  tlp_partition           INTEGER REFERENCES partition(part_number)
                            NOT FOR REPLICATION,                               -- ID of partition for this counter
  tlp_public              SMALLINT NOT NULL,                                   -- open to punlic
  site                    CHAR(32) NOT NULL,                                   -- Site Name
```

```
  tlp_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
  tlp_checksum          INTEGER NOT NULL,                              -- Record checksum
  PRIMARY KEY           ( tlp_partition, tlp_id )
);
  CREATE INDEX          tlp_public ON trigger_link_p900(tlp_public);
  CREATE UNIQUE INDEX   tlp_site_name ON trigger_link_p900 (site, tlp_name);
```

CREATE TABLE trigger_link_p900_save

```
(
  tlp_id                INTEGER NOT NULL,                              -- unique ID
  tlp_name              varchar(32),                                   -- the link name.
  tlp_enable            SMALLINT NOT NULL,                             -- enable
  tlp_source_panel      INTEGER NOT NULL,                              -- source controller
  tlp_source_event      INTEGER NOT NULL,                              -- source event
  tlp_destination_panel INTEGER NOT NULL,                              -- destination controller
  tlp_query_string      CHAR(64),                                      -- query string enter by user
  tlp_destination_event INTEGER NOT NULL,                              -- destination event
  tlp_partition         INTEGER,                                       -- ID of partition for this counter
  tlp_public            SMALLINT NOT NULL,                             -- open to punlic
  site                  CHAR(32) NOT NULL,                             -- Site Name
  tlp_guid              uniqueidentifier NOT NULL,                     -- Global unique record identifier
  tlp_checksum          INTEGER NOT NULL,                              -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                     -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                             -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                             -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                              -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,  -- Global unique record identifier
  PRIMARY KEY           ( tlp_partition, tlp_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE udfboolean

```
(
  ub_id                 INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,   -- unique ID for this UDF boolean value
  ub_cardholder_id      INTEGER REFERENCES cardholder(c_id) NOT FOR REPLICATION,  -- cardholder ID for this value
  ub_udfgen_id          INTEGER REFERENCES udfgen(ug_id) NOT FOR REPLICATION,     -- UDF gen ID for this value
  ub_boolean            SMALLINT NULL,                                 -- value
  site                  CHAR(32) NOT NULL,                             -- Site Name
  ub_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,   -- Global unique record identifier
  ub_checksum           INTEGER NOT NULL,                              -- Record checksum
  PRIMARY KEY           ( ub_id, ub_cardholder_id )
);
  CREATE NONCLUSTERED INDEX   ub_udfgen_id ON udfboolean(ub_udfgen_id ASC);
  CREATE NONCLUSTERED INDEX   ub_cardholder_id ON udfboolean(ub_cardholder_id ASC, ub_udfgen_id ASC);
```

CREATE TABLE udfboolean_save

```
(
  ub_id                 INTEGER NOT NULL,                              -- unique ID for this UDF boolean value
  ub_cardholder_id      INTEGER,                                       -- cardholder ID for this value
  ub_udfgen_id          INTEGER,                                       -- UDF gen ID for this value
  ub_boolean            SMALLINT NULL,                                 -- value
  site                  CHAR(32) NOT NULL,                             -- Site Name
  ub_guid               uniqueidentifier NOT NULL,                     -- Global unique record identifier
  ub_checksum           INTEGER NOT NULL,                              -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                     -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                             -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                             -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                              -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,  -- Global unique record identifier
  PRIMARY KEY           ( ub_id, ub_cardholder_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE udfdate

```
(
  ud_id              INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,                  -- unique ID for this UDF date value
  ud_cardholder_id   INTEGER REFERENCES cardholder(c_id) NOT FOR REPLICATION,      -- cardholder ID for this value
  ud_udfgen_id       INTEGER REFERENCES udfgen(ug_id) NOT FOR REPLICATION,         -- UDF gen ID for this value
  ud_date            DATETIME NULL,                                                -- value
  site               CHAR(32) NOT NULL,                                            -- Site Name
  ud_guid            uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                  -- Global unique record identifier
  ud_checksum        INTEGER NOT NULL,                                             -- Record checksum
  PRIMARY KEY        ( ud_id, ud_cardholder_id )
);
  CREATE NONCLUSTERED INDEX   ud_udfgen_id ON udfdate(ud_udfgen_id ASC);
  CREATE NONCLUSTERED INDEX   ud_cardholder_id ON udfdate(ud_cardholder_id ASC, ud_udfgen_id ASC);
```

CREATE TABLE udfdate_save

```
(
  ud_id              INTEGER NOT NULL,                                             -- unique ID for this UDF date value
  ud_cardholder_id   INTEGER,                                                      -- cardholder ID for this value
  ud_udfgen_id       INTEGER,                                                      -- UDF gen ID for this value
  ud_date            DATETIME NULL,                                                -- value
  site               CHAR(32) NOT NULL,                                            -- Site Name
  ud_guid            uniqueidentifier NOT NULL,                                    -- Global unique record identifier
  ud_checksum        INTEGER NOT NULL,                                             -- Record checksum
  upd_guid           uniqueidentifier NOT NULL,                                    -- Guid for update operation
  upd_mode           SMALLINT NOT NULL,                                            -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp      DATETIME NOT NULL,                                            -- Date / Time of the update
  upd_checksum       INTEGER NOT NULL,                                             -- Checksum
  upd_rowguid        uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,  -- Global unique record identifier
  PRIMARY KEY        ( ud_id, ud_cardholder_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE udfgen

```
(
  ug_id              INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,  -- unique ID for this UDF field
  ug_partition       INTEGER REFERENCES partition(part_number)
                        NOT FOR REPLICATION,                       -- partition ID to which this UDF belongs
  ug_public          SMALLINT,                                     -- 1 = public, 0 = private
  ug_type            SMALLINT,                                     -- UDF type (0 = text, 1 = number, 2 = bool, 3 = date,
                                                                      4 = selection)
  ug_label           CHAR(25) NOT NULL,                            -- UDF name
  ug_width           SMALLINT,                                     -- field width
  ug_order           SMALLINT,                                     -- UDF order
  site               CHAR(32) NOT NULL,                            -- Site Name
  ug_guid            uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,  -- Global unique record identifier
  ug_checksum        INTEGER NOT NULL,                             -- Record checksum
  ug_hidefrommis     INTEGER NOT NULL,                             -- Boolean flag to hide field from MIS
  ug_required        SMALLINT NOT NULL,                            -- 1 = value required
  PRIMARY KEY        (ug_partition, ug_id )
);
  CREATE UNIQUE INDEX   ug_part_label ON udfgen ( ug_partition, ug_label );
  CREATE UNIQUE INDEX   ug_label ON udfgen (ug_label );
```

CREATE TABLE udfgen_save

```
(
  ug_id              INTEGER NOT NULL,                             -- unique ID for this UDF field
  ug_partition       INTEGER,                                      -- partition ID to which this UDF belongs
  ug_public          SMALLINT,                                     -- 1 = public, 0 = private
  ug_type            SMALLINT,                                     -- UDF type (0 = text, 1 = number, 2 = bool, 3 = date)
  ug_label           CHAR(25) NOT NULL,                            -- UDF name
  ug_width           SMALLINT,                                     -- field width
  ug_order           SMALLINT,                                     -- UDF order
```

```
  site              CHAR(32) NOT NULL,                                  -- Site Name
  ug_guid           uniqueidentifier NOT NULL,                          -- Global unique record identifier
  ug_checksum       INTEGER NOT NULL,                                   -- Record checksum
  upd_guid          uniqueidentifier NOT NULL,                          -- Guid for update operation
  upd_mode          SMALLINT NOT NULL,                                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp     DATETIME NOT NULL,                                  -- Date / Time of the update
  upd_checksum      INTEGER NOT NULL,                                   -- Checksum
  upd_rowguid       uniqueidentifier ROWGUIDCOL DEFAULT newid()
                      UNIQUE NOT NULL,                                  -- Global unique record identifier
  ug_hidefrommis    INTEGER,                                            -- Boolean flag to hide field from MIS
  ug_required       SMALLINT NOT NULL,                                  -- 1 = value required
  PRIMARY KEY       (ug_partition, ug_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE udfnum

```
(
  un_id             INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,        -- unique ID for this UDF number value
  un_cardholder_id  INTEGER REFERENCES cardholder(c_id) NOT FOR REPLICATION, -- cardholder ID for this value
  un_udfgen_id      INTEGER REFERENCES udfgen(ug_id) NOT FOR REPLICATION, -- UDF gen ID for this value
  un_number         INTEGER NULL,                                       -- value
  site              CHAR(32) NOT NULL,                                  -- Site Name
  un_guid           uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  un_checksum       INTEGER NOT NULL,                                   -- Record checksum
  PRIMARY KEY       ( un_id, un_cardholder_id )
);
  CREATE NONCLUSTERED INDEX   un_udfgen_id ON udfnum(un_udfgen_id ASC);
  CREATE NONCLUSTERED INDEX   un_cardholder_id ON udfnum(un_cardholder_id ASC, un_udfgen_id ASC);
```

CREATE TABLE udfnum_save

```
(
  un_id             INTEGER NOT NULL,                                   -- unique ID for this UDF number value
  un_cardholder_id  INTEGER,                                            -- cardholder ID for this value
  un_udfgen_id      INTEGER,                                            -- UDF gen ID for this value
  un_number         INTEGER NULL,                                       -- value
  site              CHAR(32) NOT NULL,                                  -- Site Name
  un_guid           uniqueidentifier NOT NULL,                          -- Global unique record identifier
  un_checksum       INTEGER NOT NULL,                                   -- Record checksum
  upd_guid          uniqueidentifier NOT NULL,                          -- Guid for update operation
  upd_mode          SMALLINT NOT NULL,                                  -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp     DATETIME NOT NULL,                                  -- Date / Time of the update
  upd_checksum      INTEGER NOT NULL,                                   -- Checksum
  upd_rowguid       uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY       ( un_id, un_cardholder_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE udfselection

```
(
  us_id             INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,        -- unique ID for this UDF text value
  us_cardholder_id  INTEGER REFERENCES cardholder(c_id) NOT FOR REPLICATION, -- cardholder ID for this value
  us_udfgen_id      INTEGER REFERENCES udfgen(ug_id) NOT FOR REPLICATION, -- UDF gen ID for this value
  us_selection      CHAR(255) NULL,                                     -- value
  site              CHAR(32) NOT NULL,                                  -- Site Name
  us_guid           uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  us_checksum       INTEGER NOT NULL,                                   -- Record checksum
  PRIMARY KEY       ( us_id, us_cardholder_id )
);
  CREATE NONCLUSTERED INDEX   us_udfgen_id ON udfselection(us_udfgen_id ASC);
  CREATE NONCLUSTERED INDEX   us_cardholder_id ON udfselection(us_cardholder_id ASC, us_udfgen_id ASC);
```

CREATE TABLE udfselectionchoices

```
(
  usc_id            INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,                -- unique ID for this UDF choice
  usc_udfgen_id     INTEGER REFERENCES udfgen(ug_id) NOT FOR REPLICATION,       -- UDF gen ID for this choice
  usc_choice        CHAR(255) NOT NULL,                                         -- choice value
  site              CHAR(32) NOT NULL,                                          -- Site Name
  usc_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                -- Global unique record identifier
  usc_checksum      INTEGER NOT NULL,                                           -- Record checksum
  PRIMARY KEY       ( usc_id, usc_udfgen_id )
);
```

CREATE TABLE udfselectionchoices_save

```
(
  usc_id            INTEGER NOT NULL,                                           -- unique ID for this UDF value
  usc_udfgen_id     INTEGER,                                                    -- UDF gen ID for this value
  usc_choice        CHAR(255) NULL,                                             -- choice value
  site              CHAR(32) NOT NULL,                                          -- Site Name
  usc_guid          uniqueidentifier NOT NULL,                                  -- Global unique record identifier
  usc_checksum      INTEGER NOT NULL,                                           -- Record checksum
  upd_guid          uniqueidentifier NOT NULL,                                  -- Guid for update operation
  upd_mode          SMALLINT NOT NULL,                                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp     DATETIME NOT NULL,                                          -- Date / Time of the update
  upd_checksum      INTEGER NOT NULL,                                           -- Checksum
  upd_rowguid       uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY       ( usc_id, usc_udfgen_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE udfselection_save

```
(
  us_id             INTEGER NOT NULL,                                           -- unique ID for this UDF value
  us_cardholder_id  INTEGER,                                                    -- cardholder ID for this value
  us_udfgen_id      INTEGER,                                                    -- UDF gen ID for this value
  us_selection      CHAR(255) NULL,                                             -- value
  site              CHAR(32) NOT NULL,                                          -- Site Name
  us_guid           uniqueidentifier NOT NULL,                                  -- Global unique record identifier
  us_checksum       INTEGER NOT NULL,                                           -- Record checksum
  upd_guid          uniqueidentifier NOT NULL,                                  -- Guid for update operation
  upd_mode          SMALLINT NOT NULL,                                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp     DATETIME NOT NULL,                                          -- Date / Time of the update
  upd_checksum      INTEGER NOT NULL,                                           -- Checksum
  upd_rowguid       uniqueidentifier ROWGUIDCOL DEFAULT newid()
                        UNIQUE NOT NULL,                                        -- Global unique record identifier
  PRIMARY KEY       ( us_id, us_cardholder_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE udftext

```
(
  ut_id             INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,                -- unique ID for this UDF text value
  ut_cardholder_id  INTEGER REFERENCES cardholder(c_id) NOT FOR REPLICATION,    -- cardholder ID for this value
  ut_udfgen_id      INTEGER REFERENCES udfgen(ug_id) NOT FOR REPLICATION,       -- UDF gen ID for this value
  ut_text           CHAR(255) NULL,                                             -- value
  site              CHAR(32) NOT NULL,                                          -- Site Name
  ut_guid           uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,                -- Global unique record identifier
  ut_checksum       INTEGER NOT NULL,                                           -- Record checksum
  PRIMARY KEY       ( ut_id, ut_cardholder_id )
);
  CREATE NONCLUSTERED INDEX   ut_udfgen_id ON udftext(ut_udfgen_id ASC);
  CREATE NONCLUSTERED INDEX   ut_cardholder_id ON udftext(ut_cardholder_id ASC, ut_udfgen_id ASC);
```

CREATE TABLE udftext_save

```
(
  ut_id                 INTEGER NOT NULL,                                          -- unique ID for this UDF text value
  ut_cardholder_id      INTEGER,                                                   -- cardholder ID for this value
  ut_udfgen_id          INTEGER,                                                   -- UDF gen ID for this value
  ut_text               CHAR(255) NULL,                                            -- value
  site                  CHAR(32) NOT NULL,                                         -- Site Name
  ut_guid               uniqueidentifier NOT NULL,                                 -- Global unique record identifier
  ut_checksum           INTEGER NOT NULL,                                          -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                                 -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                         -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                          -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY           ( ut_id, ut_cardholder_id, upd_mode desc, upd_guid )
);
```

CREATE TABLE uniquebadge_id

```
(
  ubid_id               INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,               -- unique database ID
  ubid_badge_number_str VARCHAR(100) NOT NULL,                                     -- badge number
  ubid_facility_code    INTEGER NOT NULL,                                          -- facility code 0 to 65535
  ubid_issue            SMALLINT NOT NULL,                                         -- issue level
  ubid_pin_str          CHAR(20) NULL,                                             -- PIN number
  site                  CHAR(32) NOT NULL,                                         -- Site Name
  ubid_guid             uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,               -- Global unique record identifier
  ubid_checksum         INTEGER NOT NULL,                                          -- Record checksum
  PRIMARY KEY           (ubid_id)
);
  CREATE INDEX          ubid_badge_number_str ON uniquebadge_id ( ubid_badge_number_str );
  CREATE INDEX          ubid_facility_code ON uniquebadge_id ( ubid_facility_code );
  CREATE INDEX          ubid_issue ON uniquebadge_id ( ubid_issue );
```

CREATE TABLE userpartition

```
(
  up_user_id            INTEGER REFERENCES users(u_id) NOT FOR REPLICATION,        -- User ID
  up_part_number        INTEGER REFERENCES partition(part_number) NOT FOR REPLICATION, -- Partition ID
  site                  CHAR(32) NOT NULL,                                         -- Site Name
  up_guid               uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,               -- Global unique record identifier
  up_checksum           INTEGER NOT NULL,                                          -- Record checksum
  PRIMARY KEY           ( up_user_id, up_part_number)
);
```

CREATE TABLE userpartition_save

```
(
  up_user_id            INTEGER,                                                   -- User ID
  up_part_number        INTEGER,                                                   -- Partition ID
  site                  CHAR(32) NOT NULL,                                         -- Site Name
  up_guid               uniqueidentifier NOT NULL,                                 -- Global unique record identifier
  up_checksum           INTEGER NOT NULL,                                          -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                                 -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                         -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                         -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                          -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY           ( up_user_id, up_part_number, upd_mode desc, upd_guid)
);
```

CREATE TABLE userprivgrp

```
(
  upg_user_id       INTEGER REFERENCES users(u_id) NOT FOR REPLICATION,      -- user ID
  upg_group_id      INTEGER REFERENCES privgrp(pg_id) NOT FOR REPLICATION,   -- privilege group ID
  site              CHAR(32) NOT NULL,                                       -- Site Name
  upg_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,             -- Global unique record identifier
  upg_checksum      INTEGER NOT NULL,                                        -- Record checksum
  PRIMARY KEY       (upg_user_id, upg_group_id)
);
```

CREATE TABLE userprivgrp_save

```
(
  upg_user_id       INTEGER,                                                 -- user ID
  upg_group_id      INTEGER,                                                 -- privilege group ID
  site              CHAR(32) NOT NULL,                                       -- Site Name
  upg_guid          uniqueidentifier NOT NULL,                               -- Global unique record identifier
  upg_checksum      INTEGER NOT NULL,                                        -- Record checksum
  upd_guid          uniqueidentifier NOT NULL,                               -- Guid for update operation
  upd_mode          SMALLINT NOT NULL,                                       -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp     DATETIME NOT NULL,                                       -- Date / Time of the update
  upd_checksum      INTEGER NOT NULL,                                        -- Checksum
  upd_rowguid       uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY       (upg_user_id, upg_group_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE userremotepartitions

```
(
  urp_id            INTEGER IDENTITY NOT FOR REPLICATION PRIMARY KEY,        -- unique user remote partition ID
  urp_user_id       INTEGER REFERENCES users(u_id) NOT FOR REPLICATION,      -- user ID
  urp_partition_name CHAR(32) NOT NULL,                                      -- remote partition name
  site              CHAR(32) NOT NULL,                                       -- Site Name
  urp_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,             -- Global unique record identifier
  urp_checksum      INTEGER NOT NULL,                                        -- Record checksum
);
```

CREATE TABLE userremotepartitions_save

```
(
  urp_id            INTEGER NOT NULL,                                        -- unique user remote partition ID
  urp_user_id       INTEGER,                                                 -- user ID
  urp_partition_name CHAR(32) NOT NULL,                                      -- remote partition name
  site              CHAR(32) NOT NULL,                                       -- Site Name
  urp_guid          uniqueidentifier NOT NULL,                               -- Global unique record identifier
  urp_checksum      INTEGER NOT NULL,                                        -- Record checksum
  upd_guid          uniqueidentifier NOT NULL,                               -- Guid for update operation
  upd_mode          SMALLINT NOT NULL,                                       -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp     DATETIME NOT NULL,                                       -- Date / Time of the update
  upd_checksum      INTEGER NOT NULL,                                        -- Checksum
  upd_rowguid       uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY       (urp_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE users

```
(
  u_id              INTEGER IDENTITY NOT FOR REPLICATION
                      PRIMARY KEY,                                           -- unique ID for this user
  u_name            CHAR(16) NOT NULL,                                       -- login name
  u_fullname        CHAR(40) NULL,                                           -- full user name
  u_timeout         SMALLINT NULL,                                           -- auto logout timeout in mins (0 = never)
  u_password        INTEGER NULL,                                            -- encrypted password data
  u_password_1      INTEGER NULL,                                            -- encrypted password data
```

```
  u_password_2                  INTEGER NULL,                                   -- encrypted password data
  u_password_3                  INTEGER NULL,                                   -- encrypted password data
  u_addnt                       SMALLINT NOT NULL,                              -- has been added to NT account or not
  u_mfg_id                      INTEGER NULL REFERENCES msgfiltergroup(mfg_id)
                                    NOT FOR REPLICATION,                        -- message filter group ID
  u_invalid_attempt             INTEGER NOT NULL,                               -- number of consecutive invalid attempts
  u_enabled                     SMALLINT NOT NULL,                              -- account enabled
  u_pwd_never_expire            SMALLINT NOT NULL,                              -- password never expires
  u_pwd_update                  SMALLINT NOT NULL,                              -- password has to be updated on next login
  u_pwd_expiry_date             DATETIME NOT NULL,                              -- password expiry date (only valid if password
                                                                                   expires)
  u_account_mis                 SMALLINT NOT NULL,                              -- user account can be used for mis
  u_account_p2000               SMALLINT NOT NULL,                              -- user accoutn can be user for P2000 thick client
  site                          CHAR(32) NOT NULL,                              -- Site Name
  u_guid                        uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,    -- Global unique record identifier
  u_checksum                    INTEGER NOT NULL,                               -- Record checksum
  u_enable_date                 DATETIME NULL,                                  -- date when password can be renabled
  u_verify_password             SMALLINT NOT NULL,                              -- 1 requires password verification for critical apps
  u_allow_multiple_alarm_handling SMALLINT NOT NULL,                            -- Default to 1 => allow multiple; 0 => restrict to
                                                                                   one
  u_alarm_proc_grp_mfg_id       INTEGER NULL REFERENCES msgfiltergroup(mfg_id)
                                    NOT FOR REPLICATION,                        -- message filter group ID for Alarm Processing
                                                                                   Group
  u_ads_lookup                  SMALLINT NOT NULL,                              -- 1 = validate password using Active Directory
);
  CREATE UNIQUE INDEX           u_site_name ON users( site, u_name );
```

CREATE TABLE usersite

```
(
  urs_id                INTEGER IDENTITY NOT FOR REPLICATION PRIMARY KEY,   -- unique user remote partition ID
  urs_user_id           INTEGER REFERENCES users(u_id) NOT FOR REPLICATION,  -- user ID
  urs_enable            SMALLINT NOT NULL,
  site                  CHAR(32) NOT NULL,                                  -- Site Name
  urs_guid              uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,        -- Global unique record identifier
  urs_checksum          INTEGER NOT NULL,                                   -- Record checksum
);
  CREATE UNIQUE INDEX   usersite_site ON  usersite(site, urs_user_id);
  CREATE INDEX          usersite_c_id ON  usersite(urs_user_id);
```

CREATE TABLE usersite_save

```
(
  urs_id                INTEGER,                                                        -- unique user remote partition ID
  urs_user_id           INTEGER,                                                        -- user ID
  urs_enable            SMALLINT NOT NULL,
  site                  CHAR(32) NOT NULL,                                              -- Site Name
  urs_guid              uniqueidentifier NOT NULL,                                      -- Global unique record identifier
  urs_checksum          INTEGER NOT NULL,                                               -- Record checksum
  upd_guid              uniqueidentifier NOT NULL,                                      -- Guid for update operation
  upd_mode              SMALLINT NOT NULL,                                              -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp         DATETIME NOT NULL,                                              -- Date / Time of the update
  upd_checksum          INTEGER NOT NULL,                                               -- Checksum
  upd_rowguid           uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL,    -- Global unique record identifier
  PRIMARY KEY           ( urs_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE users_save

```
(
  u_id                  INTEGER NOT NULL,       -- unique ID for this user
  u_name                CHAR(16) NOT NULL,      -- login name
  u_fullname            CHAR(40) NULL,          -- full user name
  u_timeout             SMALLINT NULL,          -- auto logout timeout in mins (0 = never)
```

```
  u_password                    INTEGER NULL,                          -- encrypted password data
  u_password_1                  INTEGER NULL,                          -- encrypted password data
  u_password_2                  INTEGER NULL,                          -- encrypted password data
  u_password_3                  INTEGER NULL,                          -- encrypted password data
  u_addnt                       SMALLINT NOT NULL,                     -- has been added to NT account or not
  u_mfg_id                      INTEGER NULL,                          -- message filter group ID
  u_invalid_attempt             INTEGER NOT NULL,                      -- number of consecutive invalid attempts
  u_enabled                     SMALLINT NOT NULL,                     -- account enabled
  u_pwd_never_expire            SMALLINT NOT NULL,                     -- password never expires
  u_pwd_update                  SMALLINT NOT NULL,                     -- password has to be updated on next login
  u_pwd_expiry_date             DATETIME NOT NULL,                     -- password expiry date (only valid if password expires)
  u_account_mis                 SMALLINT NOT NULL,                     -- user account can be used for mis
  u_account_p2000               SMALLINT NOT NULL,                     -- user account can be user for P2000 thick client
  site                          CHAR(32) NOT NULL,                     -- Site Name
  u_guid                        uniqueidentifier NOT NULL,             -- Global unique record identifier
  u_checksum                    INTEGER NOT NULL,                      -- Record checksum
  upd_guid                      uniqueidentifier NOT NULL,             -- Guid for update operation
  upd_mode                      SMALLINT NOT NULL,                     -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp                 DATETIME NOT NULL,                     -- Date / Time of the update
  upd_checksum                  INTEGER NOT NULL,                      -- Checksum
  upd_rowguid                   uniqueidentifier ROWGUIDCOL DEFAULT
                                  newid() UNIQUE NOT NULL,             -- Global unique record identifier
  u_enable_date                 DATETIME NULL,                         -- date when password can be renabled
  u_verify_password             SMALLINT NOT NULL,                     -- 1 requires password verification for critical apps
  u_allow_multiple_alarm_handling SMALLINT NOT NULL,                   --Default to 1 => allow multiple; 0 => restrict to one
  u_alarm_proc_grp_mfg_id       INTEGER NULL,                          -- message filter group ID for Alarm Processing Group
  u_ads_lookup                  SMALLINT NOT NULL,                     -- 1 = validate password using Active Directory
  PRIMARY KEY                   (u_id, upd_mode desc, upd_guid)
);
```

CREATE TABLE user_concealed_udf

```
(
  ucu_id            INTEGER IDENTITY NOT FOR REPLICATION UNIQUE,       -- unique database ID
  ucu_ug_id         INTEGER REFERENCES udfgen(ug_id) NOT FOR REPLICATION, -- A udf reference
  ucu_user_id       INTEGER REFERENCES users(u_id) NOT FOR REPLICATION, -- A users reference
  site              CHAR(32) NOT NULL,                                 -- Site Name
  ucu_guid          uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL,       -- Global unique record identifier
  ucu_checksum      INTEGER NOT NULL,                                  -- Record checksum
  PRIMARY KEY       (ucu_id, site)
);
```

CREATE TABLE user_concealed_udf_save

```
(
  ucu_id            INTEGER NOT NULL,                                  -- unique identifier
  ucu_ug_id         INTEGER NOT NULL,                                  -- A udf reference
  ucu_user_id       INTEGER NOT NULL,                                  -- A users reference
  site              CHAR(32) NOT NULL,                                 -- Site Name
  ucu_guid          uniqueidentifier NOT NULL,                         -- Global unique record identifier
  ucu_checksum      INTEGER NOT NULL,                                  -- Record checksum
  upd_guid          uniqueidentifier NOT NULL,                         -- Guid for update operation
  upd_mode          SMALLINT NOT NULL,                                 -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp     DATETIME NOT NULL,                                 -- Date / Time of the update
  upd_checksum      INTEGER NOT NULL,                                  -- Checksum
  upd_rowguid       uniqueidentifier ROWGUIDCOL DEFAULT
                      newid() UNIQUE NOT NULL,                         -- Global unique record identifier
  PRIMARY KEY       (ucu_id, site, upd_mode desc, upd_guid)
);
```

CREATE TABLE webaccessconfig

```
(
  wc_mifarencode_enable   SMALLINT NOT NULL,                          -- record name
  site                    CHAR(32) NOT NULL,                          -- Site Name
  wc_guid                 uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  wc_checksum             INTEGER NOT NULL,                           -- Record checksum
  PRIMARY KEY             ( wc_guid )
)
```

CREATE TABLE webaccessconfig_save

```
(
  wc_mifarencode_enable   SMALLINT NOT NULL,                          -- record name
  site                    CHAR(32) NOT NULL,                          -- Site Name
  wc_guid                 uniqueidentifier NOT NULL,                  -- Global unique record identifier
  wc_checksum             INTEGER NOT NULL,                           -- Record checksum
  upd_guid                uniqueidentifier NOT NULL,                  -- Guid for update operation
  upd_mode                SMALLINT NOT NULL,                          -- Update mode (0 - delete, 1 - edit, 2 - new)
  upd_timestamp           DATETIME NOT NULL,                          -- Date / Time of the update
  upd_checksum            INTEGER NOT NULL,                           -- Checksum
  upd_rowguid             uniqueidentifier ROWGUIDCOL DEFAULT newid() UNIQUE NOT NULL, -- Global unique record identifier
  PRIMARY KEY             ( wc_guid, upd_mode desc, upd_guid )
)
```

CREATE TABLE webaccessctrl

```
(
  wac_id                  INTEGER IDENTITY NOT FOR REPLICATION,
  wac_cardholder_id       INTEGER REFERENCES cardholder(c_id) NOT FOR REPLICATION NOT NULL,
  wac_password_0          INTEGER NULL,
  wac_password_1          INTEGER NULL,
  wac_password_2          INTEGER NULL,
  wac_password_3          INTEGER NULL,
  site                    CHAR(32) NOT NULL,                          -- Site Name
  wac_guid                uniqueidentifier ROWGUIDCOL UNIQUE NOT NULL, -- Global unique record identifier
  wac_priv_grp_id         INTEGER NULL REFERENCES privgrp(pg_id) NOT FOR REPLICATION, -- priv group ID
  PRIMARY KEY             ( wac_id)
);
  CREATE INDEX            wac_cardholder_id ON webaccessctrl ( wac_cardholder_id );
```

CREATE TABLE xaction

```
(
  x_hist_type             INTEGER,                                    -- xaction type
                                                                        1 = Reader up, 5 = Reader down, 10 = Facility code error,
                                                                        11 = System event activated, 12 = System event deactivated,
                                                                        15 = Unlock all doors, 16 = Lock all doors, 17 = Output set,
                                                                        18 = Output reset, 19 = Reader locked, 20 = Reader unlocked,
                                                                        21 = Door held open, 22 = Door forced open,
                                                                        23 = Valid and unauthorized access, 33 = Invalid card,
                                                                        34 = Anti passback, 35 = Invalid reader, 36 = Invalid in-x-it,
                                                                        37 = Invalid card timezone, 38 = Invalid pin,
                                                                        39 = Invalid issue level, 40 = Host deny,
                                                                        42 = Invalid reader timezone, 43 = Timed override expire,
                                                                        44 = Invalid event, 45 = Invalid event privilege level,
                                                                        65 = Host grant, 67 = Executive privilege, 68 = Local grant,
                                                                        69 = Timed override enable, 70 = Timed override disable,
                                                                        71 = Timed override enable host,
                                                                        72 = Timed override disabled host, 73 = User event activated,
                                                                        74 = User event disabled, 75 = Soft in-x-it,
                                                                        78 = Manual reader, 96 = Alarm set, 97 = Alarm reset,
                                                                        99 = Tamper detected, 100 = Tamper reset,
                                                                        101 = Door open alarm, 102 = Duress alarm,
                                                                        103 = Pincode retry alarm, 104 = Forced door alarm,
```

```
                                                    106 = Prox card battery low, 107 = AC power lost,
                                                    108 = AC power restored, 109 = Low battery,
                                                    110 = Battery restored, 117 = Alarm open, 118 = Alarm short,
                                                    224 = Node up, 228 = Node down, 266 = Host grant in,
                                                    267 = Host grant out, 292 = Input terminal up,
                                                    293 = Output terminal up, 294 = Input terminal down,
                                                    295 = Output terminal down, 20481 = Node up duplicate,
                                                    20482 = Reader terminal unknown,
                                                    20483 = Input terminal unknown,
                                                    20484 = Output terminal unknown
                                                    24577 = Event triggered, 24578 = Event triggered manually
    x_panel_name            CHAR(32) NULL,          -- panel name that reported xaction (if applicable)
    x_term_name             CHAR(32) NULL,          -- terminal name involved in xaction (if applicable)
    x_item_name             CHAR(32) NULL,          -- item name involved in xaction (if applicable)
    x_badge_number          VARCHAR(100) NULL,      -- badge number involved in xaction (if applicable)
    x_fname                 CHAR(26) NULL,          -- cardholder first name involved in xaction (if applicable)
    x_lname                 CHAR(26) NULL,          -- cardholder last name involved in xaction (if applicable)
    x_timed_overrd          SMALLINT NULL,          -- override time (if timed override)
    x_issue_level           INTEGER NULL,           -- issue level of badge (if invalid issue level)
    x_fac_code              INTEGER NULL,           -- facility code of badge (if invalid facility code)
    x_event_name            CHAR(32) NULL,          -- name of event triggered (if event triggered)
    x_timestamp             DATETIME NOT NULL,      -- time/date of xaction
    site                    CHAR(32) NULL,          -- site name
    x_partition             CHAR(32) NULL,          -- partition name
    x_public                SMALLINT NOT NULL,      -- 0 = private, 1 = public
    x_guid                  uniqueidentifier DEFAULT newid() ROWGUIDCOL
                                NOT NULL,           -- unique id for this xaction
    x_checksum              INTEGER NOT NULL,       -- Record checksum
    PRIMARY KEY             NONCLUSTERED ( x_guid )
);
    CREATE CLUSTERED INDEX  x_timestamp ON xaction ( x_timestamp );
    CREATE INDEX            x_hist_type ON xaction ( x_hist_type );
    CREATE INDEX            x_site ON xaction ( site );
    CREATE INDEX            x_public ON xaction( x_public );
    CREATE INDEX            x_partition ON xaction ( x_partition );
    CREATE INDEX            x_badge_number ON xaction ( x_badge_number );
```