

**Protocol Table**

Variable	Type	Size	Null	Description
Protocol_No	Long	4	No	Protocol table unique code number
Compatibility_Code	Long	4	No	Protocol compatible code number
Method_Module	Char	32	No	Test method module name
Stimulus_Module	Char	32	No	Test stimulus module name
Extra_Module	Char	32	Yes	Extra module name
Creation_Date	Date	8	No	Protocol Creation time/date
Protocol_Flags	Long	4	No	Protocol Flags boolean bits
Category	Varchar	255	Yes	Protocol Category(s)
Description	Varchar	255	Yes	Protocol Description
Protocol_Data	Blob		Yes	Contains the actual protocol
Help_Data	Blob		Yes	Contains a help file (Not currently used)
Total_channels	Long	4	No	Number of channels used
Last_Updated	Date	8	No	Last date Protocol was updated
Deleted	ShortInt	2		Protocol has been deleted (<>0)

**Normals Table**

Variable	Type	Size	Null	Description
Normal_No	Integer	4	No	Normal table unique code number (contains serial no.)
Subject_No	Integer	4	No	Subject which contributed the normal
Test_No	Integer	4	Yes	Normal test containing just the result for this marker no.
Name	Char	16	No	Normal Marker name e.g. P100
Method_Module	Char	32	No	Test method module name
Stimulus_Module	Char	32	No	Test stimulus module name
Stimulus_Data	Varchar	255	No	Masked Stimulus Data e.g. contrast, check size etc
Sex	Char	1	Yes	M or F
Age	Integer	4	No	Age in days
Age_Range	Integer	4	No	Age range from age in days (for averages)
Pupil_Size	Integer	4	No	Pupil size in mm
Pupil_Range	Integer	4	No	Pupil size range in mm
Ethnic_Origin	Varchar	64	Yes	Ethnic origin
Electrode_Type	Varchar	16	Yes	Electrode type
Amp_Mean	Integer	4	No	Amplitude mean in nV
Amp_Deviation	Integer	4	No	Amplitude deviation in nV
Time_Mean	Integer	4	No	Time mean in us
Time_Deviation	Integer	4	No	Time deviation in us
Entered_Date	Date	8	No	Date normal entered into table
Last_Updated	Date	8	No	Last date normal was updated
Comment	Varchar	64	Yes	A comment
Total_Normals	Integer	4	No	Total normals used to create data, if this is >1 then this record is an average
Normal_Data	Blob		Yes	Contains other info (Not currently used)
Normal_Flags	Long	4	No	Protocol Flags boolean bits
Deleted	ShortInt	2	No	Normal has been deleted (<>0)

**DiagnosisCodes Table (not used)**

Variable	Type	Size	Null	Description
Diagnosis_Code	Varchar	64	No	Diagnosis code string
Description	Varchar	255	Yes	Diagnosis description

**Diagnosis Table**

Variable	Type	Size	Null	Description
Subject_No	Long	4	No	Espion subject unique number (from originating espion)

Diagnosis_Code	Char	64	No	Diagnosis code string
Diagnostist_No	Long	4	No	Diagnosis subject unique number
Diagnosis_Date	Date	8	No	Diagnosis date
Examining_Doctor	Varchar	30	No	Diagnosed by Doctor
Last_Updated	Date	8	No	Last date Protocol was updated
Deleted	ShortInt	2	No	Diagnosis has been deleted (<>0)

**Test Table**

Variable	Type	Size	Null	Description
Test_No	Long	4	No	Test table unique number
Subject_No	Long	4	No	Espion subject unique number
Protocol_No	Long	4	No	Espion Protocol unique number
Test_Date	Date	8	No	Test Date and time
Examining_Doctor	Varchar	30	No	Examining doctor
Comment	Varchar	255	Yes	A user comment about the test
Test_Data	Blob		Yes	Actual test data
Test_Flags	Long	4	Yes	Test flags Bit = 0 Set if data contains trial data
Report_Data	Blob		Yes	Might contain a report e.g. word doc (not used)
Last_Updated	Date	8	No	Last date test was updated
Normal	ShortInt	2	No	Test is normal (must be from normal subject)
Deleted	ShortInt	2	No	Test has been deleted (<>0)

**Subject Table**

Variable	Type	Size	Null	Description
Subject_No	Long	4	No	Subject table unique number
Entered_Date	Date	8	No	Date subject was first entered
Family_Name	Char	32	No	Family name
First_Name	Char	32	No	First name
DOB	Date	8	No	Date of birth
Hospital_Code	Char	32	No	Hospital number
Initial	Char	8	Yes	Initials
Title	Char	8	Yes	Title
Street	Varchar	128	Yes	Street
Town	Varchar	64	Yes	Town
State	Varchar	32	Yes	State/County
ZipCode	Varchar	16	Yes	Zipcode/Postcode
Country	Varchar	32	Yes	Country
Telephone	Varchar	32	Yes	Telephone
Sex	Char	1	No	M or F
Referring_Doctor	Varchar	32	Yes	Referring doctor
OS_Acuity	Varchar	8	Yes	OS Acuity
OD_Acuity	Varchar	8	Yes	OD Acuity
Comment	Varchar	255	Yes	A user comment
Alias_No	Long	4	Yes	Link to another subject number to show they are the same
Ethnic_Origin	Varchar	64	Yes	Country of origin/skin colour
Category	Varchar	255	Yes	Subject Category(s)
Normal	Smallint	2	No	Subject is Normal
Eye_Color	Varchar	16	Yes	Eye Color (Both eyes)
OS_Refraction	Varchar	8	Yes	Refraction in diopters
OD_Refraction	Varchar	8	Yes	Refraction in diopters
Subject_flags	Long	4	Yes	Subject flags
Last_Updated	Date	8	No	Last date Protocol was updated
Deleted	Smallint	2	No	Subject has been deleted
User_Field1	Varchar	32	Yes	User field 1
User_Field2	Varchar	32	Yes	User field 2

User_Field3	Varchar	32	Yes	User field 3
-------------	---------	----	-----	--------------

## **NOTES**

### **TABLE IDENTIFIERS**

Unique table identifier e.g. protocol\_no, diagnosis\_no, test\_no, subject\_no etc are 0 based (20 bits max) for the local machine, but as soon as they are exported, the serial number is added to produce a unique number for the entire world. This allows the database to track where the record originated from.

Total 32 bits for identifier

Save bottom 20 bits for local storage = 1million local stored table records  
 Top 12 bits for machine identification = 4096 ESPION machines maximum

The guest record is then added to the host table with the full 32 bit identifier, leaving an audit trail to record its origin

### **SUBJECT DETAILS**

When subject details are passed between Espions, the only subject details passed are:

Subject\_No as above  
 Date of Birth  
 Town  
 State  
 Sex  
 OD Acuity  
 OS Acuity  
 Ethnic Origin  
 Comment

All other fields are blank

## **GLOBAL PROCEDURES**

### **COUNTSUBJECTTESTS**

```

CREATE PROCEDURE COUNTSUBJECTTESTS (
  ISUBJECTNO INTEGER
) RETURNS (
  OCOUNT INTEGER,
  OLASTDATE DATE
) AS BEGIN

  SELECT COUNT(*),MAX(TEST_DATE) FROM TESTS WHERE SUBJECT_NO=:ISUBJECTNO INTO OCOUNT,OLASTDATE;
END

```

### **FINDPROTOCOL**

```

CREATE PROCEDURE FINDPROTOCOL ( PROTOCOLNO INTEGER)
RETURNS ( PROTOCOLCOUNT INTEGER)
AS
BEGIN
  select count(*) from protocol
  where protocol_no=:protocolno
  into :protocolcount;
END

```

### **REMOVEDELETED**

```

CREATE PROCEDURE REMOVEDELETED AS BEGIN
  DELETE FROM SUBJECTS WHERE DELETED=1;
  DELETE FROM TESTS WHERE DELETED=1;
  DELETE FROM DIAGNOSIS WHERE DELETED=1;
  DELETE FROM PROTOCOL WHERE DELETED=1;
  DELETE FROM NORMALS WHERE DELETED=1;
END

```

### **NEWDIAGNOSIS**

```

CREATE PROCEDURE NEWDIAGNOSIS (
  IDIAGNOSISNO INTEGER,
  ISERIALNO INTEGER,
  ISUBJECTNO INTEGER,
  IDIAGNOSISCODE VARCHAR(64),
  IDOCTOR VARCHAR(30)
) RETURNS (
  ODIAGNOSISNO INTEGER,
  OENTEREDDATE DATE,
  OISNEW INTEGER
) AS

BEGIN
  IF (IDIAGNOSISNO=0) then ODIAGNOSISNO = GEN_ID(DIAGNOSISNOGEN,1) + (ISERIALNO*1048576);
  ELSE ODIAGNOSISNO=IDIAGNOSISNO;

  IF (NOT EXISTS (SELECT * FROM DIAGNOSIS WHERE DIAGNOSIS_CODE=:IDIAGNOSISCODE and SUBJECT_NO=:ISUBJECTNO)) then
  BEGIN
    IF (EXISTS (SELECT SUBJECT_NO FROM SUBJECTS WHERE SUBJECT_NO=:ISUBJECTNO)) THEN
      BEGIN
        OENTEREDDATE = "NOW";
        INSERT INTO DIAGNOSIS(DIAGNOSIS_NO,DIAGNOSIS_DATE, LAST_UPDATED,DIAGNOSIS_NO,DIAGNOSIS_CODE,EXAMINING_DOCTOR,SUBJECT_NO)
        VALUES(:ODIAGNOSISNO, OENTEREDDATE, OENTEREDDATE, 0, :IDIAGNOSISCODE, :IDOCTOR, :ISUBJECTNO);
        OISNEW=1;
      END
    ELSE OISNEW=-1;
  END
  ELSE
  BEGIN
    UPDATE DIAGNOSIS SET DELETED=0,EXAMINING_DOCTOR=:IDOCTOR WHERE DIAGNOSIS_CODE=:IDIAGNOSISCODE AND SUBJECT_NO=:ISUBJECTNO AND
    DELETED<>0;
    SELECT DIAGNOSIS_NO, LAST_UPDATED FROM DIAGNOSIS WHERE DIAGNOSIS_CODE=:IDIAGNOSISCODE AND SUBJECT_NO=:ISUBJECTNO INTO
    :ODIAGNOSISNO, :OENTEREDDATE;
    OISNEW=0;
  END
END

```

### **NEWPROTOCOL**

```

CREATE PROCEDURE NEWPROTOCOL (
  IPROTOCOLNO INTEGER,
  ISERIALNO INTEGER
) RETURNS (
  OPROTOCOLNO INTEGER,
  OCREATIONDATE DATE,
  OISNEW INTEGER

```

```

) AS
BEGIN

IF (IPROTOCOLNO=0) then OPROTOCOLNO = GEN_ID(PROTOCOLNOGEN,1) + (ISERIALNO*1048576);
ELSE OPROTOCOLNO=IPROTOCOLNO;

IF (NOT EXISTS (SELECT * FROM PROTOCOL WHERE PROTOCOL_NO=:OPROTOCOLNO)) then
BEGIN
OCREATIONDATE = "NOW";
INSERT INTO
PROTOCOL(PROTOCOL_NO,COMPATIBILITY_CODE,METHOD_MODULE,STIMULUS_MODULE,CREATION_DATE,LAST_UPDATED,PROTOCOL_FLAGS,CATEGORY,
TOTAL_CHANNELS)
VALUES(:OPROTOCOLNO,0,"",:OCREATIONDATE,:OCREATIONDATE,0,"",1);
OISNEW=1;
END
ELSE
BEGIN
UPDATE PROTOCOL SET DELETED=0 WHERE PROTOCOL_NO=:OPROTOCOLNO AND DELETED<0;
SELECT LAST_UPDATED FROM PROTOCOL WHERE PROTOCOL_NO=:OPROTOCOLNO INTO :OCREATIONDATE;
OISNEW=0;
END
END
END

```

## NEWSUBJECT

```

CREATE PROCEDURE NEWSUBJECT (
ISUBJECTNO INTEGER,
ISERIALNO INTEGER
) RETURNS (
OSUBJECTNO INTEGER,
OENTEREDDATE DATE,
OISNEW INTEGER
) AS
BEGIN

IF (ISUBJECTNO=0) then OSUBJECTNO = GEN_ID(SUBJECTNOGEN,1) + (ISERIALNO*1048576);
else OSUBJECTNO=ISUBJECTNO;

IF (NOT EXISTS (SELECT * FROM SUBJECTS WHERE SUBJECT_NO=:OSUBJECTNO)) then
BEGIN
OENTEREDDATE = "NOW";
INSERT INTO SUBJECTS (SUBJECT_NO,ENTERED_DATE,LAST_UPDATED,FAMILY_NAME,FIRST_NAME,DOB,HOSPITAL_CODE,SEX,NORMAL,DELETED)
VALUES(:OSUBJECTNO,:OENTEREDDATE,:OENTEREDDATE,"",:OENTEREDDATE,0,M,0,0);
OISNEW=1;
END

ELSE
BEGIN
UPDATE SUBJECTS SET DELETED=0 WHERE SUBJECT_NO=:OSUBJECTNO AND DELETED<0;
SELECT LAST_UPDATED FROM SUBJECTS WHERE SUBJECT_NO=:OSUBJECTNO INTO :OENTEREDDATE;
OISNEW=0;
END
END
END

```

## NEWTEST

```

CREATE PROCEDURE NEWTEST (
ITESTNO INTEGER,
ISERIALNO INTEGER,
ISUBJECTNO INTEGER,
IPROTOCOLNO INTEGER,
IEXAMDOCTOR VARCHAR(30)
) RETURNS (
OTESTNO INTEGER,
OENTEREDDATE DATE,
OISNEW INTEGER
) AS
BEGIN

IF (ITESTNO=0) then OTESTNO = GEN_ID(TESTNOGEN,1) + (ISERIALNO*1048576);
else OTESTNO=ITESTNO;

IF (NOT EXISTS (SELECT * FROM TESTS WHERE TEST_NO=:OTESTNO)) then
BEGIN
IF (EXISTS (SELECT SUBJECT_NO FROM SUBJECTS WHERE SUBJECT_NO=:ISUBJECTNO)) THEN
BEGIN
IF (EXISTS (SELECT PROTOCOL_NO FROM PROTOCOL WHERE PROTOCOL_NO=:IPROTOCOLNO)) THEN
BEGIN
OENTEREDDATE = "NOW";
INSERT INTO TESTS (TEST_NO,NORMAL,TEST_DATE,LAST_UPDATED,SUBJECT_NO,PROTOCOL_NO,EXAMINING_DOCTOR,DELETED)
VALUES(:OTESTNO,0,:OENTEREDDATE,:OENTEREDDATE,:ISUBJECTNO,:IPROTOCOLNO,:IEXAMDOCTOR,0);
OISNEW=1;
END
ELSE OISNEW=-2;
END
ELSE OISNEW=-1;
END
ELSE
BEGIN
UPDATE TESTS SET DELETED=0 WHERE TEST_NO=:OTESTNO AND DELETED<0;
SELECT LAST_UPDATED FROM TESTS WHERE TEST_NO=:OTESTNO INTO :OENTEREDDATE;
OISNEW=0;
END
END
END

```

**NEWNORMAL**

```

CREATE PROCEDURE NEWNORMAL (
  INORMALNO INTEGER,
  ISERIALNO INTEGER
) RETURNS (
  ONORMALNO INTEGER,
  OENTEREDDATE DATE,
  OISNEW INTEGER
) AS
BEGIN
  IF (INORMALNO=0) then ONORMALNO = GEN_ID(NORMALNOGEN,1) + (ISERIALNO*1048576);
  ELSE ONORMALNO=INORMALNO;

  IF (NOT EXISTS (SELECT * FROM NORMALS WHERE NORMAL_NO=:ONORMALNO)) then
  BEGIN
    OENTEREDDATE = "NOW";
    INSERT INTO NORMALS
(NORMAL_NO,TEST_NO,SUBJECT_NO,NAME,METHOD_MODULE,STIMULUS_MODULE,STIMULUS_DATA,SEX,AGE,AMP_MEAN,AMP_DEVIATION,TIME_MEAN,TIME_DEVIATION,ENTERED_DATE,LAST_UPDATED,TOTAL_NORMALS,NORMAL_FLAGS,DELETED)
  VALUES(:ONORMALNO,0,0,"","","0,0,0,0,0,:OENTEREDDATE,:OENTEREDDATE,1,0,0);
    OISNEW=1;
  END
  ELSE
  BEGIN
    UPDATE NORMALS SET DELETED=0 WHERE NORMAL_NO=:ONORMALNO AND DELETED<>0;
    SELECT LAST_UPDATED FROM NORMALS WHERE NORMAL_NO=:ONORMALNO INTO :OENTEREDDATE;
    OISNEW=0;
  END
END

```

**DELETEPROTOCOL**

```

CREATE PROCEDURE DELETEPROTOCOL (
  IPROTOCOLNO INTEGER,
  ITESTSONLY INTEGER,
  IDELETED INTEGER
) AS
BEGIN
  if (IDELETED>=0) then
  begin
    UPDATE TESTS
    SET DELETED=:IDELETED
    WHERE PROTOCOL_NO=:IPROTOCOLNO;

    IF (ITESTSONLY=0) THEN
    BEGIN
      UPDATE PROTOCOL
      SET DELETED=:IDELETED
      WHERE PROTOCOL_NO=:IPROTOCOLNO;
    END
  END
  ELSE
  BEGIN
    IF (ITESTSONLY<>0) THEN DELETE FROM TESTS where PROTOCOL_NO=:IPROTOCOLNO;
    ELSE DELETE FROM PROTOCOL where PROTOCOL_NO=:IPROTOCOLNO;
  END
END

```

**DELELETSUBJECT**

```

CREATE PROCEDURE DELETESUBJECT (
  ISUBJECTNO INTEGER,
  IFLAG INTEGER,
  IDELETED INTEGER
) AS
BEGIN
  IF (IDELETED>=0) then
  BEGIN
    IF ((IFLAG=0) OR (IFLAG=1) OR (IFLAG=3) OR (IFLAG=5) OR (IFLAG=7)) THEN
    BEGIN
      UPDATE DIAGNOSIS
      SET DELETED=:IDELETED
      WHERE SUBJECT_NO=:ISUBJECTNO;
    END

    IF ((IFLAG=0) OR (IFLAG=2) OR (IFLAG=3) OR (IFLAG=6) OR (IFLAG=7)) THEN
    BEGIN
      UPDATE TESTS
      SET DELETED=:IDELETED
      WHERE SUBJECT_NO=:ISUBJECTNO;
    END

    IF ((IFLAG=0) OR (IFLAG=4) OR (IFLAG=5) OR (IFLAG=6) OR (IFLAG=7)) THEN
    BEGIN
      UPDATE NORMALS
      SET DELETED=:IDELETED
      WHERE SUBJECT_NO=:ISUBJECTNO;
    END

    IF (IFLAG=0) THEN
    BEGIN
      UPDATE SUBJECTS

```

```

SET DELETED=:IDELETED
WHERE SUBJECT_NO=:ISUBJECTNO;
END
END
ELSE
BEGIN
IF ((IFLAG=0) OR (IFLAG=1) OR (IFLAG=3) OR (IFLAG=5) OR (IFLAG=7)) THEN DELETE FROM DIAGNOSIS WHERE SUBJECT_NO=:ISUBJECTNO;
IF ((IFLAG=0) OR (IFLAG=2) OR (IFLAG=3) OR (IFLAG=6) OR (IFLAG=7)) THEN DELETE FROM TESTS WHERE SUBJECT_NO=:ISUBJECTNO;
IF ((IFLAG=0) OR (IFLAG=4) OR (IFLAG=5) OR (IFLAG=6) OR (IFLAG=7)) THEN DELETE FROM NORMALS WHERE SUBJECT_NO=:ISUBJECTNO;
IF (IFLAG=0) THEN DELETE FROM SUBJECTS where SUBJECT_NO=:ISUBJECTNO;
END
END
END

```

## DELETEDIAGNOSIS

```

CREATE PROCEDURE DELETEDIAGNOSIS (
ISUBJECTNO INTEGER,
IDIAGNOSISCODE VARCHAR(64),
IDELETED INTEGER
) AS
BEGIN
if (IDELETED>=0) then
begin
UPDATE DIAGNOSIS
SET DELETED=:IDELETED
WHERE DIAGNOSIS_CODE=:IDIAGNOSISCODE and SUBJECT_NO=:ISUBJECTNO;
END
ELSE DELETE from DIAGNOSIS where DIAGNOSIS_CODE=:IDIAGNOSISCODE and SUBJECT_NO=:ISUBJECTNO;
END
END

```

## DELETETEST

```

CREATE PROCEDURE DELETETEST (
ITESTNO INTEGER,
IFLAG INTEGER,
IDELETED INTEGER
) AS
BEGIN
IF (IDELETED>=0) then
BEGIN
IF ((IFLAG=0) or (IFLAG=1)) THEN
BEGIN
UPDATE NORMALS
SET DELETED=:IDELETED
WHERE TEST_NO=:ITESTNO;
END
END

IF (IFLAG=0) THEN
BEGIN
UPDATE TESTS
SET DELETED=:IDELETED
WHERE TEST_NO=:ITESTNO;
END
END
ELSE
BEGIN
IF (IFLAG=0) THEN DELETE from TESTS where TEST_NO=:ITESTNO;
IF ((IFLAG=0) OR (IFLAG=1)) THEN DELETE from NORMALS where TEST_NO=:ITESTNO;
END
END
END

```

## DELETENORMAL

```

CREATE PROCEDURE DELETENORMAL (
INORMALNO INTEGER,
IDELETED INTEGER
) AS
BEGIN
IF (IDELETED>=0) then
BEGIN
UPDATE NORMALS
SET DELETED=:IDELETED
WHERE NORMAL_NO=:INORMALNO;
END
ELSE DELETE from NORMALS where NORMAL_NO=:INORMALNO;
END
END

```

## REMOVEDELETED

```

CREATE PROCEDURE REMOVEDELETED AS
BEGIN
DELETE FROM SUBJECTS WHERE DELETED=1;
DELETE FROM TESTS WHERE DELETED=1;
DELETE FROM DIAGNOSIS WHERE DELETED=1;
DELETE FROM PROTOCOL WHERE DELETED=1;
DELETE FROM NORMALS WHERE DELETED=1;
END

```

**PROTOCOLDELETE**

```
CREATE TRIGGER PROTOCOLDELETE FOR PROTOCOL BEFORE DELETE POSITION 0
AS
BEGIN
DELETE FROM TESTS WHERE PROTOCOL_NO=OLD.PROTOCOL_NO;
END
```

**SUBJECTDELETE**

```
CREATE TRIGGER SUBJECTDELETE FOR SUBJECTS AFTER DELETE POSITION 0 AS
BEGIN
DELETE FROM TESTS WHERE SUBJECT_NO = OLD.SUBJECT_NO;
DELETE FROM DIAGNOSIS WHERE SUBJECT_NO = OLD.SUBJECT_NO;
END
```

**TESTDELETE**

```
CREATE TRIGGER TESTDELETE FOR TESTS AFTER DELETE POSITION 0 AS
BEGIN
DELETE FROM NORMALS WHERE TEST_NO=OLD.TEST_NO;
DELETE FROM NORMALS WHERE TEST_NO=OLD.TEST_NO;
END
```