



TimeTrak Updates to the MYOB Exo Database

TimeTrak Information
2017
Version 4.0

Table of Contents

TimeTrak SQL Scripts for MYOB Exo Integration	3
SQL Tables in MYOB Exo	4
TimeTrak Fields Added to MYOB Exo Tables	7
SQL Triggers added to MYOB Database	13
SQL Procedures and Views added or edited in MYOB Database	29
TimeTrak SQL Scripts for TimeTrak Database.....	39
Updated SQL Tables in TimeTrak.....	40
SQL Triggers in TimeTrak	77
SQL Procedures and Views added or edited in TimeTrak.....	80

TimeTrak SQL Scripts for MYOB Exo Integration

TimeTrak integrates closely with MYOB Exo Business, MYOB Job Costing and MYOB Exo CRM, so during an install and sometimes upgrades TimeTrak will add or modify tables, data and procedures in the MYOB Exo database.

To assist in keeping track of any modifications please refer to the below list of what will be altered.

On each TimeTrak Database update, the X_FocusSettings.TimeTrakVersion field will be updated to show the current version number.

SQL Tables in MYOB Exo

New Fields

Renamed Fields

Updated Fields

Removed Fields

Table is altered - fields may be removed and added

This is a list of all the SQL Database Tables, added or edited in the MYOB Exo database.

Table: X TT DisbursementGroups- This table has been altered

OLD- X DisbursementGroups

Fields:

Stockcode	varchar(23)	NOT NULL	
GroupId OLD - GID	INT	NOT NULL	

Table: X TT Settings - This table is new

Fields:

Name	NVARCHAR(50)	NOT NULL	
Value	NVARCHAR(MAX)	NOT NULL	

Table: X FocusSettings - This table has deleted

Fields:

TimeTrakVersion	varchar(5)	NOT NULL	DEFAULT '0.0'
TimeTrakExoVersion	INT	NOT NULL	DEFAULT '0.0'

Table: X TT Job Measurables Audit- This table is new

Fields:

SEQNO	INT	NOT NULL	
JOBNO	INT	NOT NULL	
POSTTIME	DATETIME	NOT NULL	GETDATE()

JOB COSTLINEID	INT	NOT NULL	0
STOCKCODE	VARCHAR(23)	NOT NULL	
DESCRIPTION	VARCHAR(60)		
PREV_MEASURE	FLOAT	NOT NULL	0
ADD_MEASURE	FLOAT	NOT NULL	0
NEW_MEASURE	FLOAT	NOT NULL	[PREV_MEASURE]+[ADD_MEASURE]
STAFFNO	INT		
DEVICEID	INT		
X_TT_TRANSID	INT	NOT NULL	0
NARRATION	VARCHAR(4096)		
JOB COSTLINESEQNO	INT	NOT NULL	0

Table: [<TIMETRAK DATABASE:Name>].dbo.ResponseLevels

OLD - X ResponseLevels- This table has been removed and added to the TimeTrak Database

Fields:

ID	INT	NOT NULL	
Description	varchar(255)	NOT NULL	
Hours	FLOAT	NOT NULL	DEFAULT '0.0'
Priority	INT	NOT NULL	0
JobType	INT	NOT NULL	0
JobCategory	INT	NOT NULL	0
ExpiryThreshold	INT	NOT NULL	0
EnableTaskExpiry	INT	NOT NULL	0
CreateInWorkHours	BIT	NOT NULL	0
Help	varchar(4096)	NOT NULL	EMPTY STRING

Table: [<TIMETRAK DATABASE:Name>].dbo.Urgencies

OLD - X XS Urgency- This table has been removed and added to the TimeTrak Database

Fields:

ID	INT	NOT NULL	
----	-----	----------	--

Description	varchar(255)	NOT NULL	
Hours	FLOAT	NOT NULL	
Help	varchar(4096)	NOT NULL	EMPTY STRING
Priority	INT	NOT NULL	1

Table: [[<TIMETRAK DATABASE:Name>](#)].dbo.UserDisbursements

OLD - X DisbursementMembership- This table has been removed and added to the TimeTrak Database

Fields:

Stockcode	varchar(23)	NOT NULL	
UID	INT	NOT NULL	
Rank	INT	NOT NULL	0
PreLoad	BIT	NOT NULL	
Quantity	FLOAT	NOT NULL	0
PurchaseOrderLine	BIT	NOT NULL	0

TimeTrak Fields Added to MYOB Exo Tables

New Fields

Renamed Fields

Updated Fields

Removed Fields

Table has been altered - fields may be removed and added

Table: Contacts- This table has been altered

Fields:

X_XS_AllowLogin	CHAR(1)	NOT NULL	'N'
X_XS_ClientAdmin	CHAR(1)	NOT NULL	'N'
X_XS_Login	varchar(50)	NOT NULL	EMPTY STRING
X_XS_Password	varchar(50)	NOT NULL	EMPTY STRING
X_XS_SendClientAdmin	CHAR(1)	NOT NULL	'N'
X_XS_ResetPassword	varchar(16)	NOT NULL	EMPTY STRING
X_XS_SortTasksBy	INT	NOT NULL	0
X_TT_CreateTasks	CHAR(1)	NOT NULL	'N'
X_TT_POContact	CHAR(1)	NOT NULL	'N'

Table: Documents

Fields:

X_TT_AttachmentId	INT	NOT NULL	0
-------------------	-----	----------	---

Table: Dr Accs- This table has been altered

Fields:

X_TT_OnlyFromTask OLD - X_OnlyFromTask	CHAR(1)	NOT NULL	'N'
X_TT_ClientApproval	CHAR(1)	NOT NULL	'N'
X_TT_ApprovalPeriod	INT		
X_TT_DefaultJobNo OLD - X_TT_DefaultJob	INT	NOT NULL	-1

X_TT_ResponseLevelId OLD - X_ResponseLevel	INT	NOT NULL	0
X_TT_LeadProviderId OLD - X_LeadProvider	INT	NOT NULL	0
X_TT_SecondaryProviderId OLD - X_SecondaryProvider	INT	NOT NULL	0
X_TT_Latitude OLD - XT_Latitude	FLOAT		
X_TT_Longitude OLD - XT_Longitude	FLOAT		
X_TT_MapZoomLevel OLD - XT_MapZoomLevel	INT	NOT NULL	14
X_TT_PortalGroupId OLD - X_TT_CustomerGroup	INT	NOT NULL	0
X_XS_AllowLogin	CHAR(1)	NOT NULL	'N'
X_TT_ReceiveDiscount	CHAR(1)	NOT NULL	'N'
X_TT_DiscountPercent	FLOAT	NOT NULL	0

Table: Dr Cont Hist - This Table has been altered

Fields:

X_TT_TimeTrakNote OLD - X_TimeTrakNote	CHAR(1)	NOT NULL	'N'
--	---------	----------	-----

Table: Job Contract Billings- This table has been altered

Fields:

X_TT_PurchOrdLineSeqNo	INT	NOT NULL	0
X_TT_InvoiceLineSeqNo	INT	NOT NULL	0

Table: JobCost Hdr- This table has been altered
Fields:

X_Description			(RTRIM([Description]))
X_TT_ContractRate OLD - X_ContractRate	FLOAT	NOT NULL	0
X_TT_IsContractRateJob OLD - X_IsContractRateJob	CHAR(1)	NOT NULL	'N'
REMOVED -X_TimeVerify1	INT	NOT NULL	-1
REMOVED -X_TimeVerify2	INT	NOT NULL	-1
REMOVED -X_Verified1	BIT	NOT NULL	0
REMOVED -X_Verified2	BIT	NOT NULL	0
X_TT_RestrictResources OLD - X_RestrictResources	CHAR	NOT NULL	'N'
X_TT_RestrictLabour OLD - X_RestrictLabour	CHAR	NOT NULL	'N'
X_TT_RestrictTaskCreation OLD - X_RestrictTaskCreation	CHAR	NOT NULL	'N'
X_TT_ClientApproval	CHAR(1)	NOT NULL	'N'
X_TT_ApprovalPeriod	INT		
X_TT_ClientVisible	CHAR(1)	NOT NULL	'Y'
X_TT_AddTasks	CHAR(1)	NOT NULL	'N'
X_TT_AutoTravel OLD - X_AutoTravel	CHAR	NOT NULL	0
X_TT_AutoTravelMins OLD - X_AutoTravelMins	INT	NOT NULL	'N'
X_TT_LeadProviderId OLD - X_LeadProvider	INT	NOT NULL	0
X_TT_SecondaryProviderId OLD - X_SecondaryProvider	INT	NOT NULL	0
X_TT_ResponseLevelId OLD - X_ResponseLevel	INT	NOT NULL	0
X_TT_Latitude OLD - XT_Latitude	FLOAT		

X_TT_Longitude OLD - XT_Longitude	FLOAT		
X_TT_MapZoomLevel OLD - XT_MapZoomLevel	INT	NOT NULL	2
X_TT_SendCopyJobManager	CHAR(1)	NOT NULL	'N'
X_TT_OnlyFromTask	CHAR(1)	NOT NULL	'N'
X_TT_TaskReadOnly	CHAR(1)	NOT NULL	'N'
X_TT_HoursVisibleToClient	CHAR(1)	NOT NULL	'Y'
X_TT_EnforceCostGroupBudget	CHAR(1)	NOT NULL	'N'
X_TT_JobTemplate OLD - X_TT_TemplateJob	CHAR(1)	NOT NULL	'N'
X_TT_EnforceCostTypeBudget	CHAR(1)	NOT NULL	'N'
X_TT_CopyQuoteLines	CHAR(1)	NOT NULL	'N'
X_TT_QuoteTemplate	CHAR(1)	NOT NULL	'N'

Table: JobCost Lines- This table has been altered

Fields:

X_TT_TaskId	INT	NOT NULL	0
X_TT_GroupId	INT		
X_TT_TaskTypeId	INT		
X_TT_CategoryId	INT		
X_TT_SuSeqNo	INT	NOT NULL	-1
X_TT_Measurable	CHAR(1)	NOT NULL	'N'
X_TT_TenderQty	FLOAT	NOT NULL	0
X_TT_TenderLineTotal	CALCULATED FIELD		
X_TT_LastMeasureDate	DATETIME		

Table: Job Transactions- This table has been altered

Fields:

REMOVED -X_Verified1	BIT	NOT NULL	0
-----------------------------	-----	----------	---

REMOVED - X_Verified2	BIT	NOT NULL	0
X_TT_OrderNo OLD -XT_OrderNo	varchar(255)	NOT NULL	EMPTY STRING
X_TT_TaskId	INT	NOT NULL	-1
X_TT_ExportId	INT	NOT NULL	0
X_TT_LinkId	INT	NOT NULL	0
X_TT_LinkType	INT	NOT NULL	0
X_TT_StartTimeUtc	datetime		
X_TT_EndTimeUtc	datetime		
X_TT_TimeZone	nvarchar(100)		
X_TT_State	INT	NOT NULL	0
X_TT_QuoteOptionId	INT	NOT NULL	0

Table: Payment Types

Fields:

X_TT_Visible	BIT	NOT NULL	0
--------------	-----	----------	---

Table: PurchOrd Hdr

Fields:

X_TT_LinkType	INT	NOT NULL	0
X_TT_LinkId	INT	NOT NULL	0

Table: PurchOrd Lines - This Table has been altered

Fields:

X_TT_NoCharge	BIT	NOT NULL	0
X_TT_QuoteOptionId	INT	NOT NULL	0

Table: Stock Items- This table has been altered

Fields:

X_CreditFromDep	INT	NULL	
-----------------	-----	------	--

X_TT_DisbursementType OLD - X_Disbursement	BIT	NOT NULL	0
X_TT_EmailPayroll OLD - X_EmailPayroll	BIT	NOT NULL	0
X_TT_IsPayrollAllowance OLD - X_Allowance	BIT	NOT NULL	0
X_TT_PayrollCode OLD - X_PayrollCode	varchar(20)	NOT NULL	EMPTY STRING
X_TT_SendToJobCosting OLD - X_SendToJobCosting	BIT	NOT NULL	0
X_TT_SendToPayrollAt OLD - X_SendToPayrollAt	FLOAT	NOT NULL	0
X_TT_SendValueToPayroll OLD - X_SendValueToPayroll	BIT	NOT NULL	0
X_TT_LineStatus OLD - X_LineStatus	INT		
X_TT_WageEasyJob OLD - X_WageEasyJob	varchar(50)	NOT NULL	EMPTY STRING
REMOVED -X_TT_TravelDisbursement	BIT	NOT NULL	0
X_TT_MinimumChargeType OLD - X_TT_MinimumCharge	INT	NOT NULL	0
X_TT_MinimumChargeDuration	INT	NOT NULL	60
X_TT_AllowZeroSellPrice OLD - X_ZeroSellPrice	BIT	NOT NULL	1
X_TT_UseCostPlusMarkup OLD - X_TT_CostPlusQuoteltem	CHAR(1)	NOT NULL	'N'
X_TT_CostPlusMarkup OLD - X_TT_Markup	FLOAT	NOT NULL	0

Table: Stock Locations- This table has been altered

Fields:

X_TT_ShortcutKey OLD - X_ShortcutKey	CHAR(1)	NULL	
--	---------	------	--

Table: Su Main- This table has been altered**Fields:**

X_TT_Latitude	FLOAT		
X_TT_Longitude	FLOAT		
X_TT_DefaultJobNo	INT	NOT NULL	0
X_TT_Address1	NVARCHAR(30)		
X_TT_Address2	NVARCHAR(30)		
X_TT_Address3	NVARCHAR(30)		
X_TT_Address4	NVARCHAR(30)		
X_TT_Address5	NVARCHAR(30)		
X_TT_Address6	NVARCHAR(30)		

SQL Triggers added to MYOB Database**New****Renamed****Updated****Removed****Item has been altered - fields may be removed and added****NEW Trigger Name:** X_TT_UpdateJob**Table:** JOBCOST_HDR**SQL Content:**

AFTER UPDATE

AS

BEGIN

-- Ignore trigger if doing a bulk update

IF @@ROWCOUNT <> 1

RETURN;

SET NOCOUNT ON;

-- Declare Variables

DECLARE @RunUpdate As bit,

```

@AccNo AS int,
@JobNo AS int,
@JobLeadProvider AS int,
@JobSecondaryProvider AS int,
@JobResponseLevel AS int,
@JobLatitude As float,
@JobLongitude As float,
@JobMapZoomLevel As int,
@ClientLeadProvider AS int,
@ClientSecondaryProvider AS int,
@ClientResponseLevel AS int,
@ClientLatitude As float,
@ClientLongitude As float,
@ClientMapZoomLevel As int;

-- Set update flag
SET @RunUpdate = 0;

-- Get job data
SELECT @JobNo = JobNo, @AccNo = AccNo, @JobLeadProvider = X_TT_LeadProviderId,
@JobSecondaryProvider = X_TT_SecondaryProviderId, @JobResponseLevel =
X_TT_ResponseLevelId, @JobLatitude = X_TT_Latitude, @JobLongitude = X_TT_Longitude,
@JobMapZoomLevel = X_TT_MapZoomLevel FROM Inserted;

-- Get client data
SELECT @ClientLeadProvider = X_TT_LeadProviderId, @ClientSecondaryProvider =
X_TT_SecondaryProviderId, @ClientResponseLevel = X_TT_ResponseLevelId, @ClientLatitude
= X_TT_Latitude, @ClientLongitude = X_TT_Longitude, @ClientMapZoomLevel =
X_TT_MapZoomLevel FROM Dr_Accs WHERE AccNo = @AccNo;

-- Check lead provider
IF UPDATE(X_TT_LeadProviderId) AND @JobLeadProvider = 0
BEGIN
SET @RunUpdate = 1;
SET @JobLeadProvider = @ClientLeadProvider;
END

-- Check secondary provider
IF UPDATE(X_TT_SecondaryProviderId) AND @JobSecondaryProvider = 0
BEGIN
SET @RunUpdate = 1;
SET @JobSecondaryProvider = @ClientSecondaryProvider;
END

```

```

-- Check response level
IF UPDATE(X_TT_ResponseLevelId) AND @JobResponseLevel = 0
BEGIN
SET @RunUpdate = 1;
SET @JobResponseLevel = @ClientResponseLevel;
END

-- Check latitude
IF UPDATE(X_TT_Latitude) AND (@JobLatitude IS NULL OR @JobLatitude = 0)
BEGIN
SET @RunUpdate = 1;
SET @JobLatitude = @ClientLatitude;
END

-- Check longitude
IF UPDATE(X_TT_Longitude) AND (@JobLongitude IS NULL OR @JobLongitude = 0)
BEGIN
SET @RunUpdate = 1;
SET @JobLongitude = @ClientLongitude;
END

-- Check map zoom level
IF UPDATE(X_TT_MapZoomLevel) AND @JobMapZoomLevel = 0
BEGIN
SET @RunUpdate = 1;
SET @JobMapZoomLevel = @ClientMapZoomLevel;
END

-- Update job details
If @RunUpdate = 1
UPDATE JobCost_Hdr SET X_TT_LeadProviderId = @JobLeadProvider,
X_TT_SecondaryProviderId = @JobSecondaryProvider, X_TT_ResponseLevelId =
@JobResponseLevel, X_TT_Latitude = @JobLatitude, X_TT_Longitude = @JobLongitude,
X_TT_MapZoomLevel = @JobMapZoomLevel WHERE JobNo = @JobNo;

END

```

Removed Trigger Name: X_TT_InsertJob

Table: JOBCOST_HDR

SQL Content:

AFTER INSERT

AS

BEGIN

```

-- Check row count
IF @@ROWCOUNT <> 1
RETURN;

SET NOCOUNT ON;

-- Declare Variables
DECLARE @RunUpdate As bit,
@AccNo AS int,
@JobNo AS int,
@JobLeadProvider AS int,
@JobSecondaryProvider AS int,
@JobResponseLevel AS int,
@JobLatitude As float,
@JobLongitude As float,
@JobMapZoomLevel As int,
@ClientLeadProvider AS int,
@ClientSecondaryProvider AS int,
@ClientResponseLevel AS int,
@ClientLatitude As float,
@ClientLongitude As float,
@ClientMapZoomLevel As int;

-- Set update flag
SET @RunUpdate = 0;

-- Get job data
SELECT @JobNo = JobNo, @AccNo = AccNo, @JobLeadProvider = X_TT_LeadProviderId,
@JobSecondaryProvider = X_TT_SecondaryProviderId, @JobResponseLevel =
X_TT_ResponseLevelId, @JobLatitude = X_TT_Latitude, @JobLongitude = X_TT_Longitude,
@JobMapZoomLevel = X_TT_MapZoomLevel FROM Inserted;

-- Get client data
SELECT @ClientLeadProvider = X_TT_LeadProviderId, @ClientSecondaryProvider =
X_TT_SecondaryProviderId, @ClientResponseLevel = X_TT_ResponseLevelId, @ClientLatitude
= X_TT_Latitude, @ClientLongitude = X_TT_Longitude, @ClientMapZoomLevel =
X_TT_MapZoomLevel FROM Dr_Accs WHERE AccNo = @AccNo;

-- Check lead provider
IF @JobLeadProvider = 0
BEGIN
SET @RunUpdate = 1;
SET @JobLeadProvider = @ClientLeadProvider;
END

```



```

-- Check secondary provider
IF @JobSecondaryProvider = 0
BEGIN
SET @RunUpdate = 1;
SET @JobSecondaryProvider = @ClientSecondaryProvider;
END

-- Check response level
IF @JobResponseLevel = 0
BEGIN
SET @RunUpdate = 1;
SET @JobResponseLevel = @ClientResponseLevel;
END

-- Check latitude
IF (@JobLatitude IS NULL OR @JobLatitude = 0)
BEGIN
SET @RunUpdate = 1;
SET @JobLatitude = @ClientLatitude;
END

-- Check longitude
IF (@JobLongitude IS NULL OR @JobLongitude = 0)
BEGIN
SET @RunUpdate = 1;
SET @JobLongitude = @ClientLongitude;
END

-- Check map zoom level
IF @JobMapZoomLevel = 0
BEGIN
SET @RunUpdate = 1;
SET @JobMapZoomLevel = @ClientMapZoomLevel;
END

-- Update job details
If @RunUpdate = 1
UPDATE JobCost_Hdr SET X_TT_LeadProviderId = @JobLeadProvider,
X_TT_SecondaryProviderId = @JobSecondaryProvider, X_TT_ResponseLevelId =
@JobResponseLevel, X_TT_Latitude = @JobLatitude, X_TT_Longitude = @JobLongitude,
X_TT_MapZoomLevel = @JobMapZoomLevel WHERE JobNo = @JobNo;

END

```

GO

NEW Trigger Name: X_CT_JOB_TRANSACTIONS_CUSTORDERNO

Table: JOB_TRANSACTIONS

SQL Content:

FOR INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON

DECLARE @JOBNO INT,

 @LINE_SOURCE INT,

 @HDR_SEQNO INT,

 @NEWCUSTORDERNO VARCHAR(20),

 @OLDCUSTORDERNO VARCHAR(20),

 @CUSTORDERNO VARCHAR(20),

 @FIRST CHAR(1),

 @ACCNO INT;

SELECT @JOBNO=JOBNO,@HDR_SEQNO=INVSEQNO FROM INSERTED WHERE INVSEQNO>0
and X_TT_ORDERNO<>"

IF @JOBNO>0

BEGIN

 SELECT @OLDCUSTORDERNO=CUSTORDERNO,@ACCNO=ACCNO FROM DR_TRANS
WHERE SEQNO=@HDR_SEQNO

 IF @ACCNO>0 AND ((@OLDCUSTORDERNO IS NULL) OR (@OLDCUSTORDERNO = ""))

 BEGIN

 SET @FIRST='Y'

 DECLARE CUSTORDERNO_CURSOR CURSOR FOR

 SELECT B.X_TT_ORDERNO FROM DR_INVLINES AS A INNER JOIN

JOB_TRANSACTIONS AS B ON A.JOBNO=B.JOBNO AND A.LINE_SOURCE=B.JOBLINEID AND
A.STOCKCODE=B.STOCKCODE WHERE A.HDR_SEQNO=@HDR_SEQNO AND B.X_TT_ORDERNO
<>"

 OPEN CUSTORDERNO_CURSOR

 FETCH NEXT FROM CUSTORDERNO_CURSOR

 INTO @CUSTORDERNO

 /* CHECK @@FETCH_STATUS TO SEE IF THERE ARE MORE ROWS TO FETCH */

 WHILE @@FETCH_STATUS=0

 BEGIN

 IF @FIRST <> 'N'

 BEGIN

 SELECT @NEWCUSTORDERNO=@CUSTORDERNO

 SET @FIRST='N'

 END ELSE

```

                SELECT
@NEWCUSTORDERNO=@NEWCUSTORDERNO+','+@CUSTORDERNO
                FETCH NEXT FROM CUSTORDERNO_CURSOR
                INTO @CUSTORDERNO
            END
            CLOSE CUSTORDERNO_CURSOR
            DEALLOCATE CUSTORDERNO_CURSOR
            IF @NEWCUSTORDERNO <> ''
            BEGIN
                UPDATE DR_TRANS
                SET CUSTORDERNO=@NEWCUSTORDERNO,REF2=@NEWCUSTORDERNO
                WHERE SEQNO=@HDR_SEQNO
            END
        END
    END
END

SET NOCOUNT OFF
END

GO

```

EDITED - Trigger Name: X_TT_ProcessPurchaseOrderLines

Table: Job_Transactions

SQL Content:

AFTER INSERT

AS

BEGIN

-- Check row count

IF @@ROWCOUNT = 0

RETURN;

SET NOCOUNT ON;

-- Update job transaction

UPDATE JT SET Line_Status = 'I', Invoiced = JT.LineTotal, InvoicedQty = JT.Quantity,

InvoiceDate = DT.TransDate, InvSeqNo = IL.Hdr_SeqNo, Invline_SeqNo = IL.SeqNo,

Schedule_SeqNo = IL.Job_Contract_Billings_SeqNo

FROM Inserted I

JOIN Job_Transactions JT ON JT.SeqNo = I.SeqNo

LEFT OUTER JOIN PurchOrd_Lines PL ON PL.PoLineId = JT.Po_LineSeqNo

LEFT OUTER JOIN Job_Contract_Billings JB ON JB.X_TT_PurchOrdLineSeqNo = PL.PoLineId

LEFT OUTER JOIN Dr_InvLines IL ON IL.SeqNo = JB.X_TT_InvoiceLineSeqNo

LEFT OUTER JOIN Dr_Trans DT ON DT.SeqNo = IL.Hdr_SeqNo

WHERE I.SeqNo = JT.SeqNo AND JT.Po_LineSeqNo > 0 AND NOT IL.Hdr_SeqNo IS NULL;

```
END
GO
```

EDITED - Trigger Name: X_TT_InsertStockItem

Table: STOCK_ITEMS

SQL Content:

```
AFTER INSERT
AS
BEGIN
-- Check row count
IF @@ROWCOUNT = 0
RETURN;

SET NOCOUNT ON;

-- Set new stock item as a disbursement
UPDATE S SET X_TT_DisbursementType = 1
FROM Inserted I
JOIN Stock_Items S ON S.StockCode = I.StockCode
WHERE I.IsActive = 'Y' AND I.Has_Sn <> 'Y';

-- Insert global group relationship
INSERT INTO X_TT_DisbursementGroups (StockCode, GroupId)
SELECT StockCode, 0 FROM Inserted
WHERE StockCode NOT IN (SELECT StockCode FROM X_TT_DisbursementGroups WHERE
GroupId = 0);

END
GO
```

```
ALTER TABLE [dbo].[Stock_Items] DISABLE TRIGGER [X_TT_InsertStockItem]
GO
```

```
/*# Message: Update X_TT_UpdateStockItem Trigger on Stock_Items Table #*/
if exists (select * FROM dbo.sysobjects where id =
OBJECT_ID(N'[dbo].[X_TT_UpdateStockItem]'))
DROP TRIGGER [dbo].[X_TT_UpdateStockItem];
GO
```

```
CREATE TRIGGER [dbo].[X_TT_UpdateStockItem] ON [dbo].[Stock_Items]
AFTER UPDATE
AS
```

```

BEGIN
-- Declare variables
DECLARE @RowCount As int,
@GroupCount As int,
@DisbursementType As int,
@OldStockCode As varchar(23),
@NewStockCode As varchar(23),
@Database As varchar(50);

-- Get row count
SET @RowCount = @@ROWCOUNT;

SET NOCOUNT ON;

-- Check if the stock code has been renamed
IF @RowCount = 1 AND UPDATE(StockCode)
BEGIN

-- Get previous stock code
SELECT @NewStockCode = StockCode FROM Inserted;
SELECT @OldStockCode = StockCode FROM Deleted;

-- Check timetrak database
SELECT @Database = Value FROM X_TT_Settings WHERE Name = 'TimeTrakDatabase';
IF @Database = '<DbItem1:Name>'
BEGIN

-- Update disbursement groups
UPDATE X_TT_DisbursementGroups SET StockCode = @NewStockCode WHERE StockCode =
@OldStockCode;

-- Update disbursement membership
UPDATE [<DbItem1:Name>].dbo.UserDisbursements SET StockCode = @NewStockCode
WHERE StockCode = @OldStockCode;

-- Update labour code membership
UPDATE [<DbItem1:Name>].dbo.LabourCodeMembership SET StockCode = @NewStockCode
WHERE StockCode = @OldStockCode;

-- Update time entry
UPDATE [<DbItem1:Name>].dbo.TimeEntry SET LabourCode = @NewStockCode WHERE
LabourCode = @OldStockCode;

-- Update disbursements

```

```
UPDATE [<DbItem1:Name>].dbo.Disbursements SET StockCode = @NewStockCode WHERE
StockCode = @OldStockCode;
```

```
-- Update checklist items
```

```
UPDATE [<DbItem1:Name>].dbo.ChecklistItems SET StockCode = @NewStockCode WHERE
StockCode = @OldStockCode;
```

```
END
```

```
END
```

```
-- Set default group for disbursement
```

```
IF @RowCount > 0 AND UPDATE(X_TT_DisbursementType)
BEGIN
```

```
-- Insert row into disbursement groups table if one doesn't already exist
```

```
INSERT INTO X_TT_DisbursementGroups
SELECT StockCode, 0 FROM Inserted WHERE X_TT_DisbursementType <> 0 AND StockCode
NOT IN (SELECT StockCode FROM X_TT_DisbursementGroups);
```

```
-- Remove row from disbursement groups table if disbursement flag is removed
```

```
DELETE FROM X_TT_DisbursementGroups WHERE StockCode IN (SELECT StockCode FROM
Inserted WHERE X_TT_DisbursementType = 0);
```

```
END
```

```
END
```

```
GO
```

EDITED - Trigger Name: X_TT_UpdateStockItem

Table: STOCK_ITEMS

SQL Content:

```
AFTER UPDATE
```

```
AS
```

```
BEGIN
```

```
-- Declare variables
```

```
DECLARE @RowCount As int,
```

```
@GroupCount As int,
```

```
@DisbursementType As int,
```

```
@OldStockCode As varchar(23),
```

```
@NewStockCode As varchar(23),
```

```
@Database As varchar(50);
```

```
-- Get row count
```

```
SET @RowCount = @@ROWCOUNT;

SET NOCOUNT ON;

-- Check if the stock code has been renamed
IF @RowCount = 1 AND UPDATE(StockCode)
BEGIN

-- Get previous stock code
SELECT @NewStockCode = StockCode FROM Inserted;
SELECT @OldStockCode = StockCode FROM Deleted;

-- Check timetrak database
SELECT @Database = Value FROM X_TT_Settings WHERE Name = 'TimeTrakDatabase';
IF @Database = 'TIMETRAK DATATBASE NAME'
BEGIN

-- Update disbursement groups
UPDATE X_TT_DisbursementGroups SET StockCode = @NewStockCode WHERE StockCode =
@OldStockCode;

-- Update disbursement membership
UPDATE [WEEKLY_TEST_TIMETRAK4].dbo.UserDisbursements SET StockCode =
@NewStockCode WHERE StockCode = @OldStockCode;

-- Update labour code membership
UPDATE [WEEKLY_TEST_TIMETRAK4].dbo.UserLabourCodes SET StockCode =
@NewStockCode WHERE StockCode = @OldStockCode;

-- Update time entry
UPDATE [WEEKLY_TEST_TIMETRAK4].dbo.TimeEntries SET LabourCode = @NewStockCode
WHERE LabourCode = @OldStockCode;

-- Update disbursements
UPDATE [WEEKLY_TEST_TIMETRAK4].dbo.Disbursements SET StockCode = @NewStockCode
WHERE StockCode = @OldStockCode;

-- Update checklist items
UPDATE [WEEKLY_TEST_TIMETRAK4].dbo.ChecklistItems SET StockCode = @NewStockCode
WHERE StockCode = @OldStockCode;

-- Update measureable transactions
UPDATE [WEEKLY_TEST_TIMETRAK4].dbo.MeasurableTransactions SET StockCode =
@NewStockCode WHERE StockCode = @OldStockCode;
```

END

END

-- Set default group for disbursement

IF @RowCount > 0 AND UPDATE(X_TT_DisbursementType)

BEGIN

-- Insert row into disbursement groups table if one doesn't already exist

INSERT INTO X_TT_DisbursementGroups

SELECT StockCode, 0 FROM Inserted WHERE X_TT_DisbursementType <> 0 AND StockCode NOT IN (SELECT StockCode FROM X_TT_DisbursementGroups);

-- Remove row from disbursement groups table if disbursement flag is removed

DELETE FROM X_TT_DisbursementGroups WHERE StockCode IN (SELECT StockCode FROM Inserted WHERE X_TT_DisbursementType = 0);

END

END

Removed - Trigger Name: X_TT_UpdateJob

Table: JOBCOST_HDR

SQL Content:

AFTER INSERT

AS

BEGIN

-- Check row count

IF @@ROWCOUNT <> 1

RETURN;

SET NOCOUNT ON;

-- Declare Variables

DECLARE @RunUpdate As bit,

@AccNo AS int,

@JobNo AS int,

@JobLeadProvider AS int,

@JobSecondaryProvider AS int,

@JobResponseLevel AS int,

@JobLatitude As float,

@JobLongitude As float,


```

@JobMapZoomLevel As int,
@ClientLeadProvider AS int,
@ClientSecondaryProvider AS int,
@ClientResponseLevel AS int,
@ClientLatitude As float,
@ClientLongitude As float,
@ClientMapZoomLevel As int;

-- Set update flag
SET @RunUpdate = 0;

-- Get job data
SELECT @JobNo = JobNo, @AccNo = AccNo, @JobLeadProvider = X_TT_LeadProviderId,
@JobSecondaryProvider = X_TT_SecondaryProviderId, @JobResponseLevel =
X_TT_ResponseLevelId, @JobLatitude = X_TT_Latitude, @JobLongitude = X_TT_Longitude,
@JobMapZoomLevel = X_TT_MapZoomLevel FROM Inserted;

-- Get client data
SELECT @ClientLeadProvider = X_TT_LeadProviderId, @ClientSecondaryProvider =
X_TT_SecondaryProviderId, @ClientResponseLevel = X_TT_ResponseLevelId, @ClientLatitude
= X_TT_Latitude, @ClientLongitude = X_TT_Longitude, @ClientMapZoomLevel =
X_TT_MapZoomLevel FROM Dr_Accs WHERE AccNo = @AccNo;

-- Check lead provider
IF UPDATE(X_TT_LeadProviderId) AND @JobLeadProvider = 0
BEGIN
SET @RunUpdate = 1;
SET @JobLeadProvider = @ClientLeadProvider;
END

-- Check secondary provider
IF UPDATE(X_TT_SecondaryProviderId) AND @JobSecondaryProvider = 0
BEGIN
SET @RunUpdate = 1;
SET @JobSecondaryProvider = @ClientSecondaryProvider;
END

-- Check response level
IF UPDATE(X_TT_ResponseLevelId) AND @JobResponseLevel = 0
BEGIN
SET @RunUpdate = 1;
SET @JobResponseLevel = @ClientResponseLevel;
END

```

```

-- Check latitude
IF UPDATE(X_TT_Latitude) AND (@JobLatitude IS NULL OR @JobLatitude = 0)
BEGIN
SET @RunUpdate = 1;
SET @JobLatitude = @ClientLatitude;
END

-- Check longitude
IF UPDATE(X_TT_Longitude) AND (@JobLongitude IS NULL OR @JobLongitude = 0)
BEGIN
SET @RunUpdate = 1;
SET @JobLongitude = @ClientLongitude;
END

-- Check map zoom level
IF UPDATE(X_TT_MapZoomLevel) AND @JobMapZoomLevel = 0
BEGIN
SET @RunUpdate = 1;
SET @JobMapZoomLevel = @ClientMapZoomLevel;
END

-- Update job details
If @RunUpdate = 1
UPDATE JobCost_Hdr SET X_TT_LeadProviderId = @JobLeadProvider,
X_TT_SecondaryProviderId = @JobSecondaryProvider, X_TT_ResponseLevelId =
@JobResponseLevel, X_TT_Latitude = @JobLatitude, X_TT_Longitude = @JobLongitude,
X_TT_MapZoomLevel = @JobMapZoomLevel WHERE JobNo = @JobNo;

END
GO

```

REMOVED - Trigger Name: X_DisbursementCheck

Table: Stock_Items

SQL Content:

```

FOR INSERT, UPDATE
AS
DECLARE @StockCode as varchar(23),
        @Disbursement as bit,
        @GroupCount as int;

SELECT @StockCode = StockCode, @Disbursement = X_Disbursement FROM Inserted;
SELECT @GroupCount = COUNT(StockCode) FROM X_DisbursementGroups WHERE StockCode =
@StockCode;

IF @Disbursement = 1

```

```

BEGIN
    IF @GroupCount = 0
        INSERT INTO X_DisbursementGroups (StockCode, GID) VALUES (@StockCode, 0)
    END
ELSE
    BEGIN
        IF @GroupCount > 0
            DELETE FROM X_DisbursementGroups WHERE StockCode = @StockCode
    END

```

REMOVED - Trigger Name: X_TT_StockCodeUpdated

Table: Stock_Items

REMOVED - Trigger Name: X_DisbursementCheck

Table: Stock_Items

REMOVED - Trigger Name: OLD - X_CT_JOB_TRANSACTIONS_CUSTORDERNO

Table: Job_Transactions

REMOVED - Trigger Name: X_TT_InsertJob

Table: JOBCOST_HDR

SQL Content:

```

AFTER INSERT
AS
BEGIN
SET NOCOUNT ON;

-- Declare Variables
DECLARE @AccNo AS int,
@JobNo AS int,
@LeadProvider AS int,
@SecondaryProvider AS int,
@ResponseLevel AS int,
@Latitude As float,
@Longitude As float,
@MapZoomLevel As int,
@ClientApproval As char(1);

-- Get Inserted Data
SELECT @JobNo = JobNo, @AccNo = AccNo, @LeadProvider = X_LeadProvider, @SecondaryProvider =
X_SecondaryProvider, @ResponseLevel = X_ResponseLevel, @Latitude = XT_Latitude, @Longitude =
XT_Longitude, @MapZoomLevel = XT_MapZoomLevel, @ClientApproval = X_TT_ClientApproval
FROM Inserted;

-- Lead Provider

```

```
IF @LeadProvider = 0
SELECT @LeadProvider = X_LeadProvider FROM Dr_Accs WHERE AccNo = @AccNo;

-- Secondary Provider
IF @SecondaryProvider = 0
SELECT @SecondaryProvider = X_SecondaryProvider FROM Dr_Accs WHERE AccNo = @AccNo;

-- Response Level
IF @ResponseLevel = 0
SELECT @ResponseLevel = X_ResponseLevel FROM Dr_Accs WHERE AccNo = @AccNo;

-- Latitude
IF (@Latitude IS NULL OR @Latitude = 0)
SELECT @Latitude = XT_Latitude FROM Dr_Accs WHERE AccNo = @AccNo;

-- Longitude
IF (@Longitude IS NULL OR @Longitude = 0)
SELECT @Longitude = XT_Longitude FROM Dr_Accs WHERE AccNo = @AccNo;

-- Map Zoom Level
IF @MapZoomLevel = 0
SELECT @MapZoomLevel = XT_MapZoomLevel FROM Dr_Accs WHERE AccNo = @AccNo;

-- Update Job Details
UPDATE JobCost_Hdr SET X_LeadProvider = @LeadProvider, X_SecondaryProvider =
@SecondaryProvider, X_ResponseLevel = @ResponseLevel, XT_Latitude = @Latitude, XT_Longitude =
@Longitude, XT_MapZoomLevel = @MapZoomLevel, X_TT_ClientApproval = @ClientApproval WHERE
JobNo = @JobNo;
```

SQL Procedures and Views added or edited in MYOB Database

New

Renamed

Updated

Removed

Item has been altered - fields may be removed and added

View Name: X_TimeTrak_JobStatusReport

SQL Content:

```

SELECT      J.JOBNO, J.JOBCODE, J.TITLE, J.QUOTEDATE, J.DUEDATE, S.STATUSDESC, S.STATUSKEY,
V1.credit, V2.timetotal, V3.other,
            SUM(T.LINETOTAL - T.INVOICED) AS Sum_Job_Transactions_Line
FROM        dbo.JOBCOST_HDR J INNER JOIN
            dbo.JOB_STATUS S ON S.STATUSKEY = J.STATUS LEFT OUTER JOIN
            dbo.JOB_TRANSACTIONS T ON T.JOBNO = J.JOBNO LEFT OUTER JOIN
            (SELECT      T .JOBNO, SUM(T .linetotal) AS credit
             FROM        JOB_TRANSACTIONS T INNER JOIN
                        STOCK_ITEMS S ON (S.STOCKCODE = T
.STOCKCODE)
             WHERE      (T .TRANSTYPE = 'C') AND (S.X_CREDITFROMDEP = 1)
             GROUP BY T .JOBNO) V1 ON J.JOBNO = V1.JOBNO LEFT OUTER JOIN
            (SELECT      T .JOBNO, SUM(T .linetotal) AS timetotal
             FROM        JOB_TRANSACTIONS T
             WHERE      (T .TRANSTYPE = 't')
             GROUP BY T .JOBNO) V2 ON J.JOBNO = V2.JOBNO LEFT OUTER JOIN
            (SELECT      T .JOBNO, SUM(T .linetotal) AS other
             FROM        JOB_TRANSACTIONS T INNER JOIN
                        STOCK_ITEMS S ON (S.STOCKCODE = T
.STOCKCODE)
             WHERE      (T .TRANSTYPE = 'C') AND ((S.X_CREDITFROMDEP <> 1)
OR
                        (S.X_CREDITFROMDEP IS NULL))
             GROUP BY T .JOBNO) V3 ON J.JOBNO = V3.JOBNO
WHERE      (J.JOBTYPE <> 21) AND (S.STATUSKEY <> 'X' OR
            S.STATUSKEY <> 'W') AND (T.TRANSTYPE <> 'Q')
GROUP BY J.JOBNO, J.JOBCODE, J.TITLE, J.QUOTEDATE, J.DUEDATE, S.STATUSDESC, S.STATUSKEY,
V1.credit, V2.timetotal, V3.other

```

REMOVED - View Name: X_TT_Groups - Moved to TimeTrak Database

SQL Content:

```

SELECT  Gid As Id, GroupName As Name FROM [<DbItem1:Name>].dbo.UserGroup WHERE IsActive = 1
AND db_name() = '<DbItem2:Name>';

```

REMOVED - View Name: X_TT_TaskTypes - Moved to TimeTrak Database**SQL Content:**

```
SELECT JobTypeId As Id, Name, IsDefault FROM [<DbItem1:Name>].dbo.TT_JobType WHERE
db_name() = '<DbItem2:Name>';
```

REMOVED - View Name: X_TT_TaskCategories - Moved to TimeTrak Database**SQL Content:**

```
SELECT CategoryId As Id, Name, IsDefault FROM [<DbItem1:Name>].dbo.TT_JobCategory WHERE
db_name() = '<DbItem2:Name>';
```

Stored Procedure Name: X_TimeTrak_AddStockTrans**SQL Content:**

```
CREATE PROCEDURE [dbo].[X_TT_AddStockTrans] (
    @StockCode AS varchar(23),
    @AccNo as int,
    @JobNo AS int,
    @Quantity AS float,
    @UnitPrice AS float,
    @UnitCost AS float,
    @Location AS int,
    @ToLocation AS int,
    @Line_SeqNo AS int,
    @KitCode AS varchar(23),
    @KitSeqNo AS int,
    @TransType AS int,
    @Ref2 AS varchar(30)
)
AS

/* Declare Local Variables */
DECLARE @Ref1 AS varchar(30),
        @JobCode AS varchar(15),
@Plu AS varchar(23),
@From_Ledger AS varchar(1),
@Post_To_GL AS varchar(1),
@UpdateItem_Flag AS varchar(1),
@BatchCode AS varchar(20),
@QtyDecimals AS int,
@SellDecimals AS int,
@CostDecimals AS int,
@Period_SeqNo AS int;

/* Look up tables to get values */
SELECT @JobCode = JobCode FROM JobCost_Hdr WHERE JobNo = @JobNo;
```

```

/* Get Decimal Places */
SELECT @QtyDecimals = FieldValue FROM Profile_Values WHERE FieldName = 'QTYDECIMALS' ORDER
BY ProfileID DESC;
If @QtyDecimals IS NULL SELECT @QtyDecimals = DefaultValue FROM Profile_Fields WHERE FieldName
= 'QTYDECIMALS';
SELECT @SellDecimals = FieldValue FROM Profile_Values WHERE FieldName = 'SELLPRDECIMALS'
ORDER BY ProfileID DESC;
If @SellDecimals IS NULL SELECT @SellDecimals = DefaultValue FROM Profile_Fields WHERE FieldName
= 'SELLPRDECIMALS';
SELECT @CostDecimals = FieldValue FROM Profile_Values WHERE FieldName = 'COSTPRDECIMALS'
ORDER BY ProfileID DESC;
If @CostDecimals IS NULL SELECT @CostDecimals = DefaultValue FROM Profile_Fields WHERE
FieldName = 'COSTPRDECIMALS';

/* Set Default Values */
SET @Ref1 = 'Job: ' + @JobCode;
SET @Plu = @StockCode;
SET @From_Ledger = 'j';
SET @Post_To_GL = 'Y';
SET @UpdateItem_Flag = 'Y';
SET @BatchCode = '';

/* Get Period Status */
SELECT TOP 1 @Period_SeqNo = SeqNo FROM Period_Status WHERE Age = 0 AND Ledger = 'S';

/* Calculate Values */
SET @Quantity = @Quantity * -1;

/* Insert the Stock Transaction */
INSERT INTO Stock_Trans (StockCode, Ref1, Ref2, Quantity, UnitPrice, UnitCost, Location, ToLocation,
Plu, From_Ledger, TransType, JobNo, Line_SeqNo, KitCode, KitSeqNo, Post_To_GL, AccNo, BatchCode,
UpdateItem_Flag, Period_SeqNo)
VALUES (@StockCode, @Ref1, @Ref2, @Quantity, @UnitPrice, @UnitCost, @Location, @ToLocation,
@Plu, @From_Ledger, @TransType, @JobNo, @Line_SeqNo, @KitCode, @KitSeqNo, @Post_To_GL,
@AccNo, @BatchCode, @UpdateItem_Flag, @Period_SeqNo);
GO

```

Stored Procedure Name: STOCK_IN_LOCATION

SQL Content:

```

ALTER PROCEDURE [dbo].[STOCK_IN_LOCATION]
    @STOCKCODE VARCHAR(23),
    @LOCATION INT,
    @ALLLOCTOTAL FLOAT = 0 OUTPUT,
    @ISINLOC FLOAT = 0 OUTPUT,
    @ISFREE FLOAT = 0 OUTPUT,

```

```

@ISCOMMITTED FLOAT = 0 OUTPUT,
@ISINCOMING FLOAT = 0 OUTPUT,
@ONBACKORDER FLOAT = 0 OUTPUT,
@ISNOTFORSALE FLOAT = 0 OUTPUT,
@NOSALELOCATION CHAR(1) = 'N' OUTPUT,
@RELEASED FLOAT = 0 OUTPUT,
@PICKED FLOAT = 0 OUTPUT,
@CORRECTED FLOAT = 0 OUTPUT,
@SUPPLIED FLOAT = 0 OUTPUT,
@UNCOSTED FLOAT = 0 OUTPUT,
@MINSTOCK FLOAT = 0 OUTPUT,
@MAXSTOCK FLOAT = 0 OUTPUT,
@BINCODE VARCHAR(12) = " OUTPUT,
@WIPLOCTOTAL FLOAT = 0 OUTPUT,
@IS_WIPLOCATION CHAR(1) = 'N' OUTPUT
AS
DECLARE @INLOC FLOAT,
        @PURCHORDS FLOAT,
        @WORDSOUT FLOAT,
        @WORDSIN FLOAT,
        @ORDS FLOAT,
        @SUP FLOAT,
        @MAXFWDDAYS INT,
        @USEWORKSORDS CHAR(1),
        @ONBACKORD FLOAT,
        @SALESORDS FLOAT,
        @BILLORDS FLOAT,
        @ONRELEASE FLOAT,
        @ONPICKED FLOAT,
        @CORRECTQUANT FLOAT,
        @ONSUPPLY FLOAT,
        @ONUNCOSTQUANT FLOAT,
        @MINSTOCKLEV FLOAT,
        @MAXSTOCKLEV FLOAT,
        @BINCODENAME VARCHAR(12),
        @PO_MAXFWDDAYS INT,
        @WO_MAXFWDDAYS INT

BEGIN
SELECT
    @ALLLOCTOTAL = ISNULL(SUM(SLI.QTY), 0),
    @WIPLOCTOTAL = ISNULL(SUM(CASE
                                WHEN SL.IS_WIPLOCATION = 'Y' THEN SLI.QTY
                                ELSE 0
                                END), 0)

FROM STOCK_LOC_INFO SLI
JOIN STOCK_LOCATIONS SL ON SLI.LOCATION = SL.LOCNO
WHERE STOCKCODE = @STOCKCODE

```



```

SELECT
  @INLOC = ISNULL(SLI.QTY, 0),
  @MINSTOCKLEV = ISNULL(SLI.MINSTOCK, 0),
  @MAXSTOCKLEV = ISNULL(SLI.MAXSTOCK, 0),
  @BINCODENAME = SLI.BINCODE,
  @IS_WIPLOCATION = SL.IS_WIPLOCATION
FROM STOCK_LOC_INFO SLI
JOIN STOCK_LOCATIONS SL ON SL.LOCNO = SLI.LOCATION AND SLI.STOCKCODE = @STOCKCODE
AND SLI.LOCATION = @LOCATION

SET @ISINLOC = @INLOC

--DEAL WITH USER PROFILES
EXECUTE GET_PROFILE_VALUE_INT 'ORDER_MAX_FWD_DAYS', @MAXFWDDAYS OUTPUT
EXECUTE GET_PROFILE_VALUE_INT 'PO_ORDER_MAX_FWD_DAYS', @PO_MAXFWDDAYS OUTPUT
EXECUTE GET_PROFILE_VALUE_INT 'WO_ORDER_MAX_FWD_DAYS', @WO_MAXFWDDAYS OUTPUT
EXECUTE GET_PROFILE_VALUE 'USEWORKSORDERS', @USEWORKSORDS OUTPUT
IF (@USEWORKSORDS IS NULL) SET @USEWORKSORDS = 'N'
IF ((@MAXFWDDAYS IS NULL) OR (@MAXFWDDAYS=0)) SET @MAXFWDDAYS = 90
IF ((@PO_MAXFWDDAYS IS NULL) OR (@PO_MAXFWDDAYS = 0)) SET @PO_MAXFWDDAYS = 90
IF ((@WO_MAXFWDDAYS IS NULL) OR (@WO_MAXFWDDAYS = 0)) SET @WO_MAXFWDDAYS = 90

IF (@USEWORKSORDS = 'Y')
BEGIN
  SELECT @WORDSOUT = ISNULL(SUM(L.QTYREQD - L.QTYUSED), 0)
  FROM WORKSORD_LINES L, WORKSORD_HDR H
  WHERE (L.HDR_SEQNO = H.SEQNO)
    AND (L.STOCKCODE = @STOCKCODE)
    AND (H.DUEDATE <= GETDATE() + @WO_MAXFWDDAYS)
    AND (H.ORDSTATUS BETWEEN 1 AND 2)
    AND H.COMPONENTLOCNO = @LOCATION

  SELECT @WORDSIN = ISNULL(SUM(PRODQTY - ACTUALQTY), 0)
  FROM WORKSORD_HDR
  WHERE (PRODCODE = @STOCKCODE)
    AND (ORDSTATUS BETWEEN 1 AND 2)
    AND (DUEDATE <= GETDATE() + @WO_MAXFWDDAYS)
    AND PRODLOCNO = @LOCATION
END
ELSE
BEGIN
  SET @WORDSOUT = 0
  SET @WORDSIN = 0
END

-- SALES ORDERS
SELECT
  @ONRELEASE = ISNULL(SUM((L.RELEASE_QUANT + L.RELEASE_NOW) * L.LINKED_QTY), 0),
  @ONPICKED = ISNULL(SUM((L.PICKED_QUANT + L.PICK_NOW) * L.LINKED_QTY), 0),

```

```

@CORRECTQUANT = ISNULL(SUM(L.CORRECTED_QUANT * L.LINKED_QTY), 0),
@SALESORDS = ISNULL(SUM(L.UNSUP_QUANT * L.LINKED_QTY), 0),
@ONBACKORD = ISNULL(SUM(L.BKORD_QUANT * L.LINKED_QTY), 0),
@ONSUPPLY = ISNULL(SUM((L.SUP_QUANT + L.SUPPLY_NOW) * L.LINKED_QTY), 0)
FROM SALESORD_LINES L
INNER JOIN SALESORD_HDR H ON L.HDR_SEQNO = H.SEQNO
WHERE (L.LINKED_STOCKCODE = @STOCKCODE)
AND ((L.HDR_STATUS < 2) OR (L.HDR_STATUS BETWEEN 4 AND 5))
AND (H.DUEDATE <= GETDATE() + @MAXFWDDAYS)
AND L.LOCATION = @LOCATION

-- PURCHASE ORDERS
SELECT
@PURCHORDS = ISNULL(SUM(((L.ORD_QUANT - L.CORRECTION_QUANT) - L.COST_QUANT) *
L.LINKED_QTY), 0),
@ONUNCOSTQUANT = ISNULL(SUM((L.SUP_QUANT - L.COST_QUANT) * L.LINKED_QTY), 0)
FROM PURCHORD_LINES L
INNER JOIN PURCHORD_HDR H ON (L.HDR_SEQNO = H.SEQNO)
WHERE (L.LINKED_STOCKCODE = @STOCKCODE)
AND (H.DUEDATE <= GETDATE() + @PO_MAXFWDDAYS)
AND (H.STATUS < 2)
AND L.LOCATION = @LOCATION

SELECT @NOSALELOCATION = EXCLUDE_FROMFREE_STOCK
FROM STOCK_LOCATIONS
WHERE LOCNO = @LOCATION

SET @ISINLOC = @INLOC
SET @ISFREE = @INLOC - @SALESORDS - @WORDSOUT + @PURCHORDS + @WORDSIN
SET @ISCOMMITTED = @SALESORDS + @WORDSOUT
SET @ISINCOMING = @PURCHORDS + @WORDSIN
SET @ONBACKORDER = @ONBACKORD
SET @RELEASED = @ONRELEASE
SET @PICKED = @ONPICKED
SET @CORRECTED = @CORRECTQUANT
SET @ISNOTFORSALE = 0

```

```

/** CUSTOM_TT **/

```

```

DECLARE @StockStatus As char;
SELECT @StockStatus = Status FROM Stock_Items WHERE StockCode = @StockCode;
IF @StockStatus <> 'L'
BEGIN
SELECT @IsNotForSale = ISNULL(SUM(Quantity),0) FROM
[<DbItem1:Name>].dbo.Disbursements D LEFT OUTER JOIN [<DbItem1:Name>].dbo.[Status] S
ON S.Sid = D.StatusId WHERE NOT D.ExportId > 0 AND S.Export = 1 AND StockCode =
@StockCode AND StockLocationId = @Location AND db_name() = '<DbItem2:Name>';
SET @IsFree = @IsFree - @IsNotForSale;
END;

```

```
/** END CUSTOM_TT **/
```

```
SET @SUPPLIED = @ONSUPPLY
SET @UNCOSTED = @ONUNCOSTQUANT
SET @MINSTOCK = @MINSTOCKLEV
SET @MAXSTOCK = @MAXSTOCKLEV
SET @BINCODE = @BINCODENAME
```

```
IF @NOSALELOCATION = 'Y'
BEGIN
    SET @ISNOTFORSALE = @ISFREE
/** CUSTOM_TT **/
IF @StockStatus <> 'L'
SET @IsNotForSale = @IsFree + @IsNotForSale;
    /** END CUSTOM_TT **/
    SET @ISFREE = 0
END
```

```
IF @IS_WIPLOCATION = 'Y'
BEGIN
    SET @ISFREE = 0
    SET @ISCOMMITTED = @INLOC
END
END
GO
```

Stored Procedure Name: STOCK_LEVELS

SQL Content:

```
ALTER PROCEDURE [dbo].[STOCK_LEVELS]
    @STOCKCODE VARCHAR(23)
AS
DECLARE
    @MAXFWDDAYS INT,
    @USEWORKSORDS CHAR(1),
    @PO_MAXFWDDAYS INT,
    @WO_MAXFWDDAYS INT
BEGIN

    DECLARE @TMP_STOCK_LEVELS TABLE
    (
        LOCNO INTEGER NOT NULL,
        ISINLOC DOUBLE PRECISION NOT NULL DEFAULT(0),
        ISFREE DOUBLE PRECISION NOT NULL DEFAULT(0),
        ISCOMMITTED DOUBLE PRECISION NOT NULL DEFAULT(0),
        ISINCOMING DOUBLE PRECISION NOT NULL DEFAULT(0),
        ONBACKORDER DOUBLE PRECISION NOT NULL DEFAULT(0),
        ISNOTFORSALE DOUBLE PRECISION NOT NULL DEFAULT(0),
```

```

RELEASED DOUBLE PRECISION NOT NULL DEFAULT(0),
PICKED DOUBLE PRECISION NOT NULL DEFAULT(0),
NOSALELOCATION CHAR(1) NOT NULL DEFAULT('N'),
CORRECTED DOUBLE PRECISION NOT NULL DEFAULT(0),
SUPPLIED DOUBLE PRECISION NOT NULL DEFAULT(0),
UNCOSTED DOUBLE PRECISION NOT NULL DEFAULT(0),
MINSTOCK DOUBLE PRECISION NOT NULL DEFAULT(0),
MAXSTOCK DOUBLE PRECISION NOT NULL DEFAULT(0),
BINCODE VARCHAR(12),
WOUT DOUBLE PRECISION NOT NULL DEFAULT(0),
WIN DOUBLE PRECISION NOT NULL DEFAULT(0),
SO DOUBLE PRECISION NOT NULL DEFAULT(0),
PO DOUBLE PRECISION NOT NULL DEFAULT(0),
IS_WIPLOCATION CHAR(1) NOT NULL DEFAULT('N'),
TIMETRAK DOUBLE PRECISION NOT NULL DEFAULT(0) /** CUSTOM_TT **/
)

-- PREPOPULATE WITH STUFF WHICH DOESNT REQUIRED AGGREGATION
INSERT INTO @TMP_STOCK_LEVELS(
    LOCNO, ISINLOC, MINSTOCK, MAXSTOCK, BINCODE, NOSALELOCATION, IS_WIPLOCATION)
SELECT L.LOCNO, ISNULL(I.QTY, 0), ISNULL(I.MINSTOCK, 0), ISNULL(I.MAXSTOCK, 0),
    I.BINCODE, L.EXCLUDE_FROMFREE_STOCK, L.IS_WIPLOCATION
FROM STOCK_LOC_INFO I
JOIN STOCK_LOCATIONS L ON I.LOCATION = L.LOCNO
WHERE STOCKCODE = @STOCKCODE

--DEAL WITH USER PROFILES
EXECUTE GET_PROFILE_VALUE_INT 'ORDER_MAX_FWD_DAYS', @MAXFWDDAYS OUTPUT
EXECUTE GET_PROFILE_VALUE_INT 'PO_ORDER_MAX_FWD_DAYS', @PO_MAXFWDDAYS OUTPUT
EXECUTE GET_PROFILE_VALUE_INT 'WO_ORDER_MAX_FWD_DAYS', @WO_MAXFWDDAYS OUTPUT
EXECUTE GET_PROFILE_VALUE 'USEWORKSORDERS', @USEWORKSORDS OUTPUT
IF (@USEWORKSORDS IS NULL) SET @USEWORKSORDS = 'N'
IF ((@MAXFWDDAYS IS NULL) OR (@MAXFWDDAYS=0)) SET @MAXFWDDAYS = 90
IF ((@PO_MAXFWDDAYS IS NULL) OR (@PO_MAXFWDDAYS=0)) SET @PO_MAXFWDDAYS = 90
IF ((@WO_MAXFWDDAYS IS NULL) OR (@WO_MAXFWDDAYS=0)) SET @WO_MAXFWDDAYS = 90

-- UPDATE RESULTS WITH AGGREGATED VALUES
IF (@USEWORKSORDS = 'Y')
BEGIN
    UPDATE T SET
        WOUT = SUBQ.QTY
    FROM @TMP_STOCK_LEVELS T
    INNER JOIN (
        SELECT H.COMPONENTLOCNO AS LOCNO, ISNULL(SUM(L.QTYREQD - L.QTYUSED), 0) AS QTY
        FROM WORKSORD_LINES L, WORKSORD_HDR H
        WHERE (L.HDR_SEQNO = H.SEQNO)
            AND (L.STOCKCODE = @STOCKCODE)
            AND (H.DUEDATE <= GETDATE() + @WO_MAXFWDDAYS)
    )

```

```

AND (H.ORDSTATUS BETWEEN 1 AND 2)
GROUP BY H.COMPONENTLOCNO) AS SUBQ ON T.LOCNO = SUBQ.LOCNO

```

```

UPDATE T SET
  WIN = SUBQ.QTY
FROM @TMP_STOCK_LEVELS T
INNER JOIN (
  SELECT PRODLOCNO AS LOCNO, ISNULL(SUM(PRODQTY - ACTUALQTY), 0) AS QTY
  FROM WORKSORD_HDR
  WHERE (PRODCODE = @STOCKCODE)
  AND (ORDSTATUS BETWEEN 1 AND 2)
  AND (DUEDATE <= GETDATE() + @WO_MAXFWDDAYS)
  GROUP BY PRODLOCNO) AS SUBQ ON T.LOCNO = SUBQ.LOCNO

```

```

END
ELSE --NO USE WORK ORDERS
BEGIN
  UPDATE @TMP_STOCK_LEVELS
  SET WOUT = 0,
  WIN = 0
END

```

```

UPDATE T SET
  RELEASED = SUBQ.RELEASED,
  PICKED = SUBQ.PICKED,
  CORRECTED = SUBQ.CORRECTED,
  SO = SUBQ.SO,
  ONBACKORDER = SUBQ.ONBACKORDER,
  SUPPLIED = SUBQ.SUPPLIED
FROM @TMP_STOCK_LEVELS T
INNER JOIN (
  SELECT  L.LOCATION AS LOCNO,
  RELEASED = ISNULL(SUM((L.RELEASE_QUANT + L.RELEASE_NOW) * L.LINKED_QTY), 0),
  PICKED = ISNULL(SUM((L.PICKED_QUANT + L.PICK_NOW) * L.LINKED_QTY), 0),
  CORRECTED = ISNULL(SUM(L.CORRECTED_QUANT * L.LINKED_QTY), 0),
  SO = ISNULL(SUM(L.UNSUP_QUANT * L.LINKED_QTY), 0),
  ONBACKORDER = ISNULL(SUM(L.BKORD_QUANT * L.LINKED_QTY), 0),
  SUPPLIED = ISNULL(SUM((L.SUP_QUANT + L.SUPPLY_NOW) * L.LINKED_QTY), 0)
  FROM SALESORD_LINES L
  INNER JOIN SALESORD_HDR H ON L.HDR_SEQNO = H.SEQNO
  WHERE  (L.LINKED_STOCKCODE = @STOCKCODE)
  AND ((L.HDR_STATUS < 2) OR (L.HDR_STATUS BETWEEN 4 AND 5))
  AND (H.DUEDATE <= GETDATE() + @MAXFWDDAYS)
  GROUP BY L.LOCATION) AS SUBQ ON T.LOCNO = SUBQ.LOCNO

```

```

UPDATE T SET
  PO = SUBQ.PO,
  UNCOSTED = SUBQ.UNCOSTED
FROM @TMP_STOCK_LEVELS T
INNER JOIN (

```

```

SELECT L.LOCATION AS LOCNO,
       PO = ISNULL(SUM(((L.ORD_QUANT - L.CORRECTION_QUANT) - L.COST_QUANT) *
L.LINKED_QTY), 0),
       UNCOSTED = ISNULL(SUM((L.SUP_QUANT - L.COST_QUANT) * L.LINKED_QTY), 0)
FROM PURCHORD_LINES L
INNER JOIN PURCHORD_HDR H ON (L.HDR_SEQNO = H.SEQNO)
WHERE (L.LINKED_STOCKCODE = @STOCKCODE)
      AND (H.DUEDATE <= GETDATE() + @PO_MAXFWDDAYS)
      AND (H.STATUS < 2)
GROUP BY L.LOCATION ) AS SUBQ ON T.LOCNO = SUBQ.LOCNO

```

```

/** CUSTOM_TT **/

```

```

UPDATE T SET T.TimeTrak = DQ.Quantity
FROM @Tmp_Stock_Levels T
INNER JOIN

```

```

(SELECT D.StockLocationId As LocNo, ISNULL(SUM(D.Quantity), 0) As Quantity FROM
[<DbItem1:Name>].dbo.Disbursements D LEFT OUTER JOIN [<DbItem1:Name>].dbo.[Status] S
ON S.Sid = D.StatusId WHERE NOT D.ExportId > 0 AND S.Export = 1 AND StockCode =
@StockCode AND db_name() = '<DbItem2:Name>' GROUP BY D.StockLocationId) AS DQ ON
T.LocNo = DQ.LocNo

```

```

/** END CUSTOM_TT **/

```

```

SELECT
LOCNO, ISINLOC,
CASE
  WHEN NOSALELOCATION = 'Y' OR IS_WIPLICATION = 'Y' THEN 0
  ELSE ISINLOC - SO - WOUT + PO + WIN - TIMETRAK /** CUSTOM_TT **/
END AS ISFREE,
CASE
  WHEN IS_WIPLICATION = 'Y' THEN ISINLOC
  ELSE (SO + WOUT)
END AS ISCOMMITTED,
(PO + WIN) AS ISINCOMING,
ONBACKORDER,
CASE
  WHEN NOSALELOCATION = 'Y' OR IS_WIPLICATION = 'Y' THEN ISINLOC + SO - WOUT + PO +
WIN
  ELSE TIMETRAK /** CUSTOM_TT **/
END AS ISNOTFORSALE,
RELEASED, PICKED, NOSALELOCATION, CORRECTED, SUPPLIED, UNCOSTED, MINSTOCK,
MAXSTOCK, BINCODE
FROM @TMP_STOCK_LEVELS
END
GO

```

TimeTrak SQL Scripts for TimeTrak Database

For TimeTrak 4.0 we have made many database changes, which include renaming tables and fields. This will have an impact on any customisation a client may have including any custom reports.

An upgrade will need to be scoped and quoted.

To assist in keeping track of any modifications please refer to the below list of what will be altered.

On each TimeTrak Database update, the SettingValues.Version field will be updated to show the current version number.

Updated SQL Tables in TimeTrak

New Fields

Renamed

Updated Fields

Removed Fields

Table has been altered - fields may be removed and added

This is a list of all the SQL Database Tables which are edited in the TimeTrak database.

Table:Activities - **This table has been renamed and updated**

OLD- Activity

Fields:

Application	NVARCHAR(100)	NOT NULL	''
Version	NVARCHAR(20)	NOT NULL	''
Latitude	FLOAT		
Longitude	FLOAT		
Altitude	FLOAT		
Accuracy	FLOAT		
AltitudeAccuracy	FLOAT		
Heading	FLOAT		
Speed	FLOAT		

Table:AdministratorContacts - **This table has been renamed and updated**

OLD- AdministratorContact

Fields:

AID OLD-ID	INT	NOT NULL	1
----------------------	-----	----------	---

Table:Appointments - **This table is updated**

Fields:

TimeEntryStatusId OLD - StatusId	INT	NOT NULL	
TimeEntryLabourCode OLD - LabourCode	NVARCHAR(23)		
TimeEntryCostGroupId OLD - CostGroupId	INT		
TimeEntryCostTypeId OLD - CostTypeId	INT		
TimeEntryRate OLD - Rate	FLOAT		
LastModifiedById OLD - LastModifiedBy	INT	NOT NULL	
CreatedById OLD - CreatedBy	INT	NOT NULL	
TimeEntryDefaults	BIT	NOT NULL	0
TimeEntryStatusId	INT		
TimeEntryLabourCode	NVARCHAR(23)		
Locked	BIT	NOT NULL	0
LockedById	INT	NOT NULL	0
LockedDate	DATETIME		
StatusId	INT	NOT NULL	0
CategoryId	INT	NOT NULL	0
PurchaseOrderId	INT	NOT NULL	0
RelatedGuid	NVARCHAR(50)	NOT NULL	''
JobId	INT	NOT NULL	0
Duration	FLOAT		DATEDIFF(minute, [StartTime], [EndTime]), 0) / 60
TimeEntryQuoteOptionId	INT		
TimeEntryServiceableUnitId	INT		
QueueItemId	INT	NOT NULL	0
RecurrenceId	INT	NOT NULL	0
CheckInGuid	NVARCHAR(50)		

Duration	CALCULATED FIELD		
----------	------------------	--	--

Table:AppointmentCategories- This table is new

Fields:

ID	INT	NOT NULL	
Name	NVARCHAR(100)	NOT NULL	
Colour	NVARCHAR(7)	NOT NULL	
ISACTIVE	BIT	NOT NULL	

Table:AppointmentRecurrences- This table is new

Fields:

Id	INT	NOT NULL	
BaseDate	DATETIME	NOT NULL	
RecurrenceInfo	NVARCHAR(Max)	NOT NULL	
RelatedGuid	NVARCHAR(50)	NOT NULL	
NonWorkdayBehaviour	INT	NOT NULL	
OnCompletionNotifyEmailTo	NVARCHAR(Max)		

Table:AppointmentRequestUsers- This table is new

Fields:

RequestId	INT	NOT NULL	
UserId	INT	NOT NULL	
State	INT	NOT NULL	

Table:AppointmentServiceableUnits- This table is new

Fields:

AppointmentId	INT	NOT NULL	
ServiceableUnitId	INT	NOT NULL	

Table:AppointmentStatuses- This table is new**Fields:**

ID	INT	NOT NULL	
Name	NVARCHAR(100)	NOT NULL	
Colour	NVARCHAR(7)	NOT NULL	
ISACTIVE	BIT	NOT NULL	

Table:Approvals- This table is renamed and updated**OLD: Approval****Fields:**

Id OLD - ApprovalId	INT	NOT NULL	
SentById OLD - SentBy	INT	NOT NULL	

Table:Attachments - This table is updated**Fields:**

Description	NVARCHAR(max) OLD - NVARCHAR(4096)	NOT NULL	
LinkType OLD - AttachmentType	INT	NOT NULL	
CreatedById OLD - CreatedBy	INT	NOT NULL	
CreatedDate OLD - DateCreated	DATETIME	NOT NULL	
LastModifiedDate OLD - LastModified	DATETIME	NOT NULL	
LastModifiedById OLD - LastModifiedBy	INT	NOT NULL	
ContentLength	BIGINT	NOT NULL	0
StorageType	INT	NOT NULL	

ContentGuid	NVARCHAR(50)	NOT NULL	''
CategoryId	INT	NOT NULL	0

Table: AttachmentFiles - This table is new

Fields:

Id	INT	NOT NULL	
FileContents	VARBINARY(MAX)	NOT NULL	

Table: AttachmentCategories - This table is new

Fields:

ID	INT	NOT NULL	1
Name	NVARCHAR(100)	NOT NULL	''

Table: BOMMembership - This table is Removed

Fields:

UID	INT	NOT NULL	
BillCode	VARCHAR(23)	NOT NULL	
IsDefault	BIT	NOT NULL	

Table: Breaks - This table is updated

Fields:

Id OLD - BID	INT	NOT NULL	
UserId OLD - UID	INT	NOT NULL	

Table: CheckIns - This table is updated

Fields:

TimeRemaining	INT	NOT NULL	
RelatedGuid	NVARCHAR(50)	NOT NULL	
RecurrenceIndex	INT	NOT NULL	-1

Table: ChecklistEvents- This table is new

Fields:

Id	INT	NOT NULL	1
ChecklistId	INT	NOT NULL	
ProfileId	INT	NOT NULL	
EventType	INT	NOT NULL	
SaveBehaviour	INT	NOT NULL	1
Rank	INT	NOT NULL	

Table: ChecklistGroups- This table is new

Fields:

Id	INT	NOT NULL	
Name	NVARCHAR(50)	NOT NULL	
Description	NVARCHAR(Max)	NOT NULL	
ChecklistId	INT	NOT NULL	

Table: ChecklistItemOptions- This table is new

Fields:

Id	INT	NOT NULL	1
ChecklistItemId	INT	NOT NULL	
Caption	NVARCHAR(100)	NOT NULL	
Value	NVARCHAR(Max)	NOT NULL	
PartialChecklistId	INT	NOT NULL	1
EmailType	INT	NOT NULL	
EmailTo	NVARCHAR(Max)	NOT NULL	
Rank	INT	NOT NULL	

Table: ChecklistJobCategories- This table is new**Fields:**

ChecklistId	INT	NOT NULL	
JobCategoryNo	INT	NOT NULL	

Table: ChecklistJobStatuses- This table is new**Fields:**

ChecklistId	INT	NOT NULL	
JobStatusKey	CHAR(1)	NOT NULL	

Table: ChecklistJobTypes- This table is new**Fields:**

ChecklistId	INT	NOT NULL	
JobTypeId	INT	NOT NULL	

Table: ChecklistReports- This table is new**Fields:**

Id	INT	NOT NULL	1
ChecklistId	INT	NOT NULL	
ReportId	INT	NOT NULL	
EmailTo	NVARCHAR(MAX)	NOT NULL	
EmailType	INT	NOT NULL	
TemplateId	INT	NOT NULL	
CategoryId	INT	NOT NULL	
CanDisable	BIT	NOT NULL	
ClientVisible	BIT	NOT NULL	

Table: ChecklistResultItems - This table is updated**Fields:**

ChecklistItemParentId	INT	NOT NULL	
-----------------------	-----	----------	--

GroupId	INT	NOT NULL	0
Rank	INT	NOT NULL	0
ChecklistItemId	INT	NOT NULL	0
DisbursementId	INT	NOT NULL	0
ServiceableUnitId	INT	NOT NULL	0
SignatureName	NVARCHAR(100)	NOT NULL	''
SignatureImage	NVARCHAR(Max)	NOT NULL	''
Uniqueld	nvarchar(255)	NOT NULL	''
AttachmentId	INT	NOT NULL	0
Label	NVARCHAR(Max)	NOT NULL	
Visibility			
DataType	INT	NOT NULL	
DataValue	NVARCHAR(Max)	NOT NULL	
SecondaryValue	INT	NOT NULL	''
DataType	INT	NOT NULL	0
DataValue	NVARCHAR(MAX)	NOT NULL	

Table: ChecklistResults - This table is updated

Fields:

JobId OLD - JobNo	INT	NOT NULL	0
Name	NVARCHAR(250)	NOT NULL	
CreatedById OLD - UserId	INT	NOT NULL	
ChecklistId	INT	NOT NULL	0
JobId	INT	NOT NULL	0
EventType OLD - ChecklistEvent	INT	NOT NULL	
State	INT	NOT NULL	0
CreatedDate OLD - ChecklistDate	DATETIME	NOT NULL	

LastModifiedById	INT	NOT NULL	0
LastModifiedDate	DATETIME	NOT NULL	getdate()
AttachmentId	INT	NOT NULL	0
Latitude	FLOAT		
Longitude	FLOAT		
Accuracy	FLOAT		

Table: Checklists - This table is updated

Fields:

ShowRecipients	BIT	NOT NULL	0
ToggleRecipients	BIT	NOT NULL	0
ChecklistGuid	NVARCHAR(50)	NOT NULL	''
Type	INT	NOT NULL	0
EmbedTemplateInPdf	BIT	NOT NULL	
TemplateId	INT	NOT NULL	
ReportId	INT	NOT NULL	
EmailTo	NVARCHAR(250)	NOT NULL	
EmailJobContact	BIT	NOT NULL	
EmailJobManager	BIT	NOT NULL	
EmailTaskContact	BIT	NOT NULL	
EmailType			
AttachmentCategoryId			

Table: ChecklistItems - This table is updated

Fields:

Visibility			
PartialChecklistId OLD - ConditionalChecklistId	INT	NOT NULL	0
ExcludeWhenBlank	BIT	NOT NULL	0
LimitToJobType	INT	NOT NULL	

LimitToJobCategory	INT	NOT NULL	
EmailTo	NVARCHAR(max)	NOT NULL	
ParentId	INT	NOT NULL	
DataSourceType	INT	NOT NULL	0

Table: ClockOns - This table is new

Fields:

ID	INT	NOT NULL	1
UserId	INT	NOT NULL	0
ClockOnDate	DATE	NOT NULL	
ClockedOnAt	TIME	NOT NULL	0
ClockedOnById	INT	NOT NULL	
ClockOnNote	NVARCHAR(500)	NOT NULL	
ClockedOffAt	DATETIME		
ClockedOffById	INT	NOT NULL	0
ClockOffNote	NVARCHAR(500)	NOT NULL	
RelatedGuid]	NVARCHAR(50)	NOT NULL	

Table: CostTypes - This table is updated

OLD - CostType

Fields:

Id OLD - CTID	INT	NOT NULL	
Name OLD - Description			
ApplyPriceToDisbursements OLD - ApplyToDisbursements			

Table: CostTypeGroups - This table is updated

OLD - CostTypeGroup

Fields:

CostTypeId OLD - CTID	INT	NOT NULL	
GroupId OLD - Gid			

Table: CustomFields - This table is updated

Fields:

LookupTable	NVARCHAR(MAX)	NOT NULL	
-------------	---------------	----------	--

Table: Disbursements - This table is updated

Fields:

ServiceableUnitId	INT	NOT NULL	
QuoteLineSeqNo	INT	NOT NULL	0
CostPrice	FLOAT	NOT NULL	0
Description	VARCHAR(60)	NOT NULL	''
JobId	INT	NOT NULL	0
Type	INT	NOT NULL	
QuoteLineId	INT	NOT NULL	0
QuoteOptionId	INT	NOT NULL	
Verified	BIT	NOT NULL	0
CreatedBy OLD - CreatedBy	INT	NOT NULL	
QuoteLineId OLD - QuoteLineSeqNo	INT	NOT NULL	

Table: ExoStatuses - This table is updated

Old - ExonetStatus

Fields:

StatusId OLD - SID			
------------------------------	--	--	--

Table: Exports - This table is updated**Old - Export****Fields:**

Id OLD - ExportID			
ExportedById OLD - ExportedBy			

Table: FieldNames- This table is updated**Fields:**

Past	NVARCHAR(50)	NOT NULL	
Present	NVARCHAR(50)	NOT NULL	

Table: Groups- This table is renamed and updated**OLD - UserGroup****Fields:**

Id OLD - GID	INT	NOT NULL	
Name OLD - GroupName	NVARCHAR(100)	NOT NULL	
JobResourceId OLD - ResourceId	INT	NOT NULL	
JobTypeId OLD - DefaultJobType	INT	NOT NULL	
Name	NVARCHAR(100)	NOT NULL	
EmailAddress	NVARCHAR(100)	NOT NULL	

Table: GroupEmailUsers- This table is updated**Fields:**

GroupId OLD - GID	INT	NOT NULL	
UserId OLD - UID	INT	NOT NULL	

Table: Holidays- This table is updated

Fields:

Description	NVARCHAR(MAX)	NOT NULL	
Id OLD - HolidayID	INT	NOT NULL	
IsActive OLD - Active			

Table: KbAnswers- This table is new

Fields:

Id	INT	NOT NULL	
ArticleId	INT	NOT NULL	
Description	NVARCHAR(Max)	NOT NULL	
PostedById	INT	NOT NULL	
PostedByClient	BIT	NOT NULL	
PostedDate	DATETIME	NOT NULL	
Accepted	BIT	NOT NULL	

Table: KbArticleCategories- This table is new

Fields:

ArticleId	INT	NOT NULL	
CategoryId	INT	NOT NULL	

Table: KbArticles- This table is new

Fields:

Id	INT	NOT NULL	
----	-----	----------	--

GroupId	INT	NOT NULL	
Title	NVARCHAR(500)	NOT NULL	
Description	NVARCHAR(Max)	NOT NULL	
ArticleType	INT	NOT NULL	
State	INT	NOT NULL	
AllowComments	BIT	NOT NULL	
Visibility	INT	NOT NULL	
CreatedById	INT	NOT NULL	
CreatedByClient	BIT	NOT NULL	
CreatedDate	DATETIME	NOT NULL	
LastModifiedById	INT	NOT NULL	
LastModifiedByClient	BIT	NOT NULL	
LastModifiedDate	DATETIME	NOT NULL	
LastResponseDate	DATETIME	NOT NULL	
IsActive	BIT	NOT NULL	

Table:KbAttachments- This table is new

Fields:

Id	INT	NOT NULL	
Name	NVARCHAR(100)	NOT NULL	
Description	NVARCHAR(Max)	NOT NULL	
CreatedById	INT	NOT NULL	
CreatedByClient	BIT	NOT NULL	
CreatedDate	datetime2	NOT NULL	
FileData	VARBINARY(Max)	NOT NULL	
LinkId	INT	NOT NULL	
LinkType	INT	NOT NULL	

Table:KbCategories- This table is new

Fields:

Id	INT	NOT NULL	
----	-----	----------	--

Name	NVARCHAR(50)	NOT NULL	
GroupId	INT	NOT NULL	
ParentId	INT	NOT NULL	

Table:KbComments- This table is new

Fields:

Id	INT	NOT NULL	
Description	NVARCHAR(Max)	NOT NULL	
PostedById	INT	NOT NULL	
PostedByClient	BIT	NOT NULL	
PostedDate	DATETIME2	NOT NULL	
LinkId	INT	NOT NULL	
LinkType	INT	NOT NULL	

Table:KbFavourites- This table is new

Fields:

ArticleId	INT	NOT NULL	
UserId	INT	NOT NULL	
ContactId	INT	NOT NULL	

Table:KbGroups- This table is new

Fields:

Id	INT	NOT NULL	
Name	NVARCHAR(100)	NOT NULL	
AllowedTypes	INT	NOT NULL	
IsActive	BIT	NOT NULL	

Table:KbGroupPermissions- This table is new

Fields:

GroupId	INT	NOT NULL	
---------	-----	----------	--

ProfileId	INT	NOT NULL	
ReadOnly	BIT	NOT NULL	
Visible	BIT	NOT NULL	

Table:KbRatings- This table is new

Fields:

UserId	INT	NOT NULL	
ContactId	INT	NOT NULL	
LinkId	INT	NOT NULL	
LinkType	INT	NOT NULL	
Value	INT	NOT NULL	

Table:KbRelatedItems- This table is new

Fields:

ArticleId	INT	NOT NULL	
LinkId	NVARCHAR(50)	NOT NULL	
LinkType	INT	NOT NULL	
JobId	INT	NOT NULL	

Table:KbSubscriptions- This table is new

Fields:

UserId	INT	NOT NULL	
LinkId	INT	NOT NULL	
LinkType	INT	NOT NULL	
Articles	INT	NOT NULL	
Questions	INT	NOT NULL	

Table: MeasurableTransactions- This table is new

Fields:

ID	INT	NOT NULL	1
QuoteLineSeqNo	INT	NOT NULL	

JobNo	INT	NOT NULL	
UserID	INT	NOT NULL	
Stockcode	VARCHAR(23)	NOT NULL	
Description	VARCHAR(60)		
PrevMeasure	FLOAT	NOT NULL	
Quantity	FLOAT	NOT NULL	
TransDate	DATETIME	NOT NULL	
Narration	VARCHAR(4096)		
CreatedByID	INT	NOT NULL	
CreatedDate	DATETIME	NOT NULL	

Table: Narrations - This table is updated

OLD -Narration

Fields:

Narration	NVARCHAR(max) OLD -NVARCHAR(4096)	NOT NULL	
Id OLD - NID			
Name	NVARCHAR(100)	NOT NULL	

Table: NarrationGroups - This table is updated

OLD -NarrationGroup

Fields:

NarrationId OLD - Nid		NOT NULL	
GroupId OLD - Gid			

Table: PortalGroups - This table is renamed and updated

OLD - TC CustomerGroups

Fields:

Name	NVARCHAR(50) OLD - VARCHAR(50)	NOT NULL	
EnableTaskVisibility	BIT	NOT NULL	
AddTasks	BIT	NOT NULL	
EditTasks	BIT	NOT NULL	
TaskEmailTo	VARCHAR(100)	NOT NULL	
TaskAssignedTo	INT	NOT NULL	
TaskGroup	INT	NOT NULL	
TaskStatus	INT	NOT NULL	
ShowJob	BIT	NOT NULL	
ShowContact	BIT	NOT NULL	
ShowTaskId	BIT	NOT NULL	
ShowAge	BIT	NOT NULL	
ShowDueDate	BIT	NOT NULL	
ShowDateCreated	BIT	NOT NULL	
ShowEstimatedTime	BIT	NOT NULL	
ShowStatus	BIT	NOT NULL	
ShowUrgency	BIT	NOT NULL	
ShowAssignedTo	BIT	NOT NULL	
ShowOrderNo	BIT	NOT NULL	
ShowTaskCategory	BIT	NOT NULL	
ShowTaskType	BIT	NOT NULL	
OverrideNewTaskLayout	BIT	NOT NULL	
OverrideEditTaskLayout	BIT	NOT NULL	
TaskUrgency	INT	NOT NULL	
UploadAttachments	BIT	NOT NULL	

Table: PortalWizardFields- This table is renamed and updated
OLD - TC TaskFields
Fields:

Help	NVARCHAR(max) OLD - NVARCHAR(4096)	NOT NULL	
CustomFieldId	INT	NOT NULL	0

Table: PortalWizardPages - **This table is renamed and updated**
OLD - TC WizardPages

Fields:

PortalGroupId OLD - CustomerGroupId	INT	NOT NULL	
Name	NVARCHAR(50) OLD - VARCHAR(50)	NOT NULL	
Description	NVARCHAR(max) OLD - NVARCHAR(2000)	NOT NULL	

Table: PortalTaskGridFields - **This table is updated**

Fields:

HorizontalAlign	INT		
CssClass	NVARCHAR(100)	NOT NULL	''
FormatString	NVARCHAR(100)	NOT NULL	''
PortalGroupId OLD - CustomerGroupId	INT	NOT NULL	

Table: PortalWizardFields - **This table is updated**

Fields:

CustomFieldId	INT	NOT NULL
---------------	-----	----------

Table: Queues - **This table is new**

Fields:

ID	INT	NOT NULL
ProfileId	INT	NOT NULL
Name	NVARCHAR(50)	NOT NULL

Rank	INT	NOT NULL
UpdateJobStatus	CHAR(1)	
UpdateJobCategory	INT	NOT NULL
UpdateJobManager	INT	NOT NULL
UpdateJobAppointmentStatus	INT	NOT NULL
UpdateJobAppointmentCategory	INT	NOT NULL
UpdateTaskStatusId	INT	NOT NULL
UpdateTaskUser	INT	NOT NULL
UpdateTaskAppointmentStatus	INT	NOT NULL
UpdateTaskAppointmentCategory	INT	NOT NULL
RemoveAfterScheduled	BIT	NOT NULL
Latitude	FLOAT	
Longitude	FLOAT	
MapZoomLevel	INT	NOT NULL
PopulateSql	NVARCHAR(MAX)	
PopulateName]	NVARCHAR(50)	
Colour	NVARCHAR(7)	NOT NULL
ScheduleBy	INT	NOT NULL
SchedulingColumns	INT	NOT NULL
DefaultDuration	INT	NOT NULL
UpdateQueueItemQueueId	INT	NOT NULL
State	INT	NOT NULL
ClearOnPopulate	BIT	NOT NULL
ItemLimit	INT	NOT NULL
PopulateInterval	INT	NOT NULL
LastPopulated	DATETIME	

Table: QueueItems- This table is new

Fields:

Id	INT	NOT NULL
QueueId	INT	NOT NULL
LinkType	INT	NOT NULL

LinkId	INT	NOT NULL
State	INT	NOT NULL
Rank	INT	NOT NULL
Selected	BIT	NOT NULL
LastModifiedById	INT	NOT NULL
ErrorMessage	NVARCHAR(100)	

Table:Reports- This table is updated

Fields:

NAME OLD - ReportName	NVARCHAR(50)	NOT NULL
Visibility OLD - Application	INT	NOT NULL
Enabled	bit	
Parameters	NVARCHAR(Max)	NOT NULL
CategoryId	INT	NOT NULL
ImportGuid	NVARCHAR(50)	NOT NULL
ImportVersion	INT	NOT NULL
SourceType	INT	NOT NULL

Table:ReportCategories- This table is new

Fields:

ID	INT	NOT NULL
NAME	NVARCHAR(100)	NOT NULL
Rank	INT	NOT NULL

Table:ReportFiles- This table is new

Fields:

ID	INT	NOT NULL
FileContents	VARBINARY(MAX)	NOT NULL

Table:ReportPermissions- This table is new**Fields:**

ReportId	INT	NOT NULL
ProfileId	INT	NOT NULL

Table:ReportServers- This table is new**Fields:**

Id	INT	NOT NULL
Name	NVARCHAR(100)	NOT NULL
Url	NVARCHAR(MAX)	NOT NULL
PublishPath	NVARCHAR(MAX)	NOT NULL
ServerGuid	NVARCHAR(MAX)	NOT NULL

Table:ResponseLevels- This table is new**Fields:**

Id	INT	NOT NULL
Name	NVARCHAR(255)	NOT NULL
Hours	FLOAT	NOT NULL
ExpiryThreshold	INT	NOT NULL
EnableTaskExpiry	BIT	NOT NULL
PriorityId	INT	NOT NULL
TaskTypeId	INT	NOT NULL
TaskCategoryId	INT	NOT NULL
CreateInWorkHours	BIT	NOT NULL
Help	NVARCHAR(MAX)	NOT NULL

Table:Rosters- This table is new**Fields:**

ID	INT	NOT NULL
NAME	NVARCHAR(100)	NOT NULL

Table:RosterClasses- This table is updated**Fields:**

ShortCode	NVARCHAR(1)	NOT NULL
PayrollCode	NVARCHAR(20)	NOT NULL

Table:RosterDays- This table is new**Fields:**

RosterID	INT	NOT NULL
DayOfWeek	INT	NOT NULL
DayStart	TIME	NOT NULL
DayEnd	TIME	NOT NULL
DailyMinimum	FLOAT	NOT NULL
WorkDay	BIT	NOT NULL

Table:RosterDayUnits- This table is new**Fields:**

ID	INT	NOT NULL	
RosterId	INT	NOT NULL	
DayOfWeek	INT	NOT NULL	
TimeOfDay	TIME	NOT NULL	
RosterClassId	INT	NOT NULL	

Table: Status- This table is updated**Fields:**

IncludeInBudget	BIT	NOT NULL	0
IncludeInProdCalc OLD - IncludeProdCalc	BIT	NOT NULL	
ID OLD - SID			

Name OLD - Status			
BranchId OLD - BranchNo			

Table: TaskCategories - This table is updated and renamed

OLD - TT JobCategory

Fields:

IsActive	BIT	NOT NULL	1
ID OLD - CategoryID	INT	NOT NULL	
NAME	NVARCHAR(100) OLD - Varchar(50)	NOT NULL	

Table: TaskDependencies - This table is new

Fields:

ID	INT	NOT NULL
ParentId	INT	NOT NULL
DependentId	INT	NOT NULL
Type	INT	NOT NULL

Table: TaskLogs - This table is updated

Old - TT TaskLog

Fields:

Id OLD - LogID		
ResponseToId OLD - ResponseTo		
CreatedById OLD - CreatedBy		
CreatedByClient OLD - ClientCreated		

LastModifiedById OLD - LastModifiedBy		
---	--	--

Table: TaskRecurrences- This table is updated

Old - TT Recurrence

Fields:

Id OLD - RID		
NextTaskGroupId	INT	NOT NULL

Table: TaskServiceableUnits- This table is new

Fields:

TaskID	INT	NOT NULL	
ServiceableUnitId	INT	NOT NULL	''

Table: TaskStatuses- This table is updated and renamed

OLD - TT Status

Fields:

ID OLD - StatusID	INT	NOT NULL
NAME	NVARCHAR(100) OLD - Varchar(50)	NOT NULL
IsComplete	BIT	NOT NULL
MustChange	BIT	NOT NULL
ClientTasksVisible	BIT	NOT NULL
IsActive	BIT	NOT NULL
OrderIndex	INT	NOT NULL
IsComplete	BIT	NOT NULL

Table: TaskTypes- This table is updated and renamed

OLD - TT JobType

Fields:

IsActive	BIT	NOT NULL	1
ID OLD - JobTypeID	INT	NOT NULL	
NAME	NVARCHAR(100) OLD - Varchar(50)	NOT NULL	

Table: TaskTypeQuestions - This table is updated**OLD - TT Questions****Fields:**

Source	NVARCHAR(max) OLD - NVARCHAR(2000)	NOT NULL	
TaskTypeID OLD - JobTypeID			
ParentID OLD - ParentQuestion			
OrderIndex OLD - QuestionOrder			

Table: Tasks - This table is updated**OLD - TT Tasks****Fields:**

Description	NVARCHAR(max) OLD - NVARCHAR(4096)	NOT NULL	
Notes	NVARCHAR(MAX) OLD - NVARCHAR(4096)	NOT NULL	
RelatedGuid	NVARCHAR(50)	NOT NULL	''
Locked	BIT	NOT NULL	0
LockedById	INT	NOT NULL	0
LockedDate	DATETIME		

BillingQuery	NVARCHAR(255)	NOT NULL	''
Baseld	INT	NOT NULL	0
ClientVisible	BIT	NOT NULL	0
Latitude	FLOAT		
Longitude	FLOAT		
UserID OLD - UID	INT	NOT NULL	
Id OLD - TaskID			
JobID OLD - JobNo			
EstimatedTime OLD - TimeRequired			
TotalFromSubTasks OLD - TotalSubTask			
ActualComplete OLD - PercentComplete			
PriorityId OLD - Priority			
StatusId OLD - Status			
TypeId OLD - jobType			
CategoryId OLD - jobCategory			
UrgencyId OLD - Urgency			
ContactId OLD - contact			
CreatedById OLD - CreatedBy			
CreatedByClient OLD - ClientCreated			

CreatedDate OLD - DateEntered			
LastModifiedById OLD - LastModifiedBy			
AssignedById OLD - AssignedBy			
ResponseLevelId OLD - ResponseLevel			
SyncToOutlook OLD - SyncTask			
RecurrenceId OLD - Rid			
GroupId OLD - GID	INT	NOT NULL	
BillableStatusId OLD - Sid			

Table: Templates- This table is updated and renamed

OLD - EmailTemplates

Fields:

Name OLD - Description	INT	NOT NULL
NAME	NVARCHAR(100) OLD - Varchar(50)	NOT NULL
Subject	NVARCHAR(500)	NOT NULL
Message	NVARCHAR(MAX)	NOT NULL

Category	INT	NOT NULL
----------	-----	----------

Table: TimeAnalyses - This table is updated and renamed
OLD - NonProductive

Fields:

Id OLD - NPID	INT	NOT NULL
StatusId OLD - SID		
Name OLD - Detail		
IncludeInProdCalc OLD - IncludeProdCalc		
NAME	NVARCHAR(50) OLD - Varchar(50)	NOT NULL

Table: TimeAnalysisGroups - This table is updated and renamed
OLD - NonProductiveGroup

Fields:

TimeAnalysisId OLD - NPID	INT	NOT NULL
GroupId OLD - GID		

Table: TaskTypeQuestions - This table is updated and renamed
OLD - TT Questions

Fields:

TaskTypeID OLD - JobTypeID	INT	NOT NULL
NAME	NVARCHAR(50) OLD - Varchar(50)	NOT NULL
LabelText	NVARCHAR(100) OLD - Varchar(100)	NOT NULL

ValidationText	NVARCHAR(Max) OLD - Varchar(1000)	NOT NULL
ValidationRegex	NVARCHAR(Max) OLD - Varchar(1000)	NOT NULL
Help	NVARCHAR(Max) OLD - Varchar(4096)	NOT NULL

Table: TimeEntries - This table is updated
OLD - TimeEntry
Fields:

Narration	NVARCHAR(max) OLD - NVARCHAR(4096)	NOT NULL	
ServiceableUnitSeqNo	INT	NOT NULL	
BillCode	VARCHAR(23)	NOT NULL	
RelatedGuid	NVARCHAR(50)	NOT NULL	''
CreatedByld	INT	NOT NULL	0
CreatedDate	DATETIME	NOT NULL	getdate()
Notes	NVARCHAR(MAX)	NOT NULL	''
SplitLineId	INT	NOT NULL	0
Duration	FLOAT		DATEDIFF(minute, [StartTime], [EndTime]), 0) / 60
QuoteOptionId	INT	NOT NULL	
TimeInLieu	FLOAT	NOT NULL	
Duration	FLOAT	NOT NULL	
USERID OLD - UID	INT		
StatusId OLD - Sid			
CostGroupId OLD - CostGroup			
CostTypeId OLD - CostType			

ApprovalSentAt OLD - ApprovalSent			
ApprovalState OLD - Approval			
ApprovedAt OLD - Approved			
TimeAnalysisId OLD - NPID	INT		
JobId OLD - JobNo			
Verified OLD - Checked			
PurchaseOrderId OLD - PurchaseOrderSeqNo			

Table: TimeEntryServiceableUnits - This table is new

Fields:

EntryId	INT	NOT NULL
ServiceableUnitId	INT	NOT NULL

Table: Urgencies - This table is new

Fields:

Id	INT	NOT NULL
Name	NVARCHAR(255)	NOT NULL
Hours	FLOAT	NOT NULL
PriorityId	INT	NOT NULL
Help	NVARCHAR(MAX)	

Table: TT_TaskLog - This table is updated

Fields:

Description	NVARCHAR(max) OLD - NVARCHAR(4096)	NOT NULL
-------------	--	----------

Table: Users- This table is renamed and updated
OLD - ApplicationUser

Fields:

CalendarClick	INT	NOT NULL	Now in SettingsFields and SettingsValues Tables
CalendarType	INT	NOT NULL	Now in SettingsFields and SettingsValues Tables
DateView	INT	NOT NULL	Now in SettingsFields and SettingsValues Tables
DaysOnCalendar	INT	NOT NULL	Now in SettingsFields and SettingsValues Tables
DefaultTaskSortField	INT	NOT NULL	Now in SettingsFields and SettingsValues Tables
DefaultTaskSortOrder	INT	NOT NULL	Now in SettingsFields and SettingsValues Tables
ExpandLevel	INT	NOT NULL	Now in SettingsFields and SettingsValues Tables
LoginMessage	VARCHAR(4096)	NOT NULL	Now in SettingsFields and SettingsValues Tables
TaskEmail	BIT	NOT NULL	Now in SettingsFields and SettingsValues Tables

TaskTrakView	INT	NOT NULL	Now in SettingsFields and SettingsValues Tables
TaskView	INT	NOT NULL	Now in SettingsFields and SettingsValues Tables
TimeFormat	INT	NOT NULL	Now in SettingsFields and SettingsValues Tables
WorkInProgress	INT	NOT NULL	Now in SettingsFields and SettingsValues Tables
ShowRate	BIT	NOT NULL	Now in SettingsFields and SettingsValues Tables
CalendarGroupBy	INT	NOT NULL	Now in SettingsFields and SettingsValues Tables
CalendarShowGroups	INT	NOT NULL	Now in SettingsFields and SettingsValues Tables
ClosedTasksVisible	BIT	NOT NULL	Now in SettingsFields and SettingsValues Tables
NewTaskAssignment	INT	NOT NULL	Now in SettingsFields and SettingsValues Tables
PreLoadTaskTitle	BIT	NOT NULL	Now in SettingsFields and SettingsValues Tables
RefreshMins	INT	NOT NULL	Now in SettingsFields and SettingsValues Tables
SharedCalendars	VARCHAR(200)	NOT NULL	Now in SettingsFields and SettingsValues Tables

ShowAcceptanceTab	BIT	NOT NULL	Now in SettingsFields and SettingsValues Tables
ShowAllSubTasks	BIT	NOT NULL	Now in SettingsFields and SettingsValues Tables
TaskAcceptedEmail	BIT	NOT NULL	Now in SettingsFields and SettingsValues Tables
OverwriteSetup	BIT	NOT NULL	
ID OLD - UID	INT	NOT NULL	
JobResourceId OLD - StaffID	INT	NOT NULL	
Name OLD - Username	NVARCHAR(100) OLD - VARCHAR(100)	NOT NULL	
WageCost OLD - LatestCost	FLOAT	NOT NULL	
StockLocationId OLD - StockLocation	INT	NOT NULL	
BranchId OLD - BranchNo			
SupplierId OLD - AccNo			
CostTypeId OLD - CostTypeSeqNo			
LoginId	NVARCHAR(20) OLD - VARCHAR(20)		
Email	NVARCHAR(100) OLD - VARCHAR(100)		

Password	NVARCHAR(100) OLD - VARCHAR(100)		
OutlookProfile	NVARCHAR(100) OLD - VARCHAR(50)		
PayrollCode	NVARCHAR(50) OLD - VARCHAR(50)		
BasePayRateCode	NVARCHAR(50) OLD - VARCHAR(20)		
LocationName	NVARCHAR(50) OLD - VARCHAR(50)		
Style	NVARCHAR(50) OLD - VARCHAR(50)		
CreatedById	INT	NOT NULL	
CreatedDate	DATETIME	NOT NULL	getdate()
StartDate	DATETIME	NOT NULL	getdate()
RosterId	INT	NOT NULL	0
ShowOnMap	BIT	NOT NULL	1
AccNo	INT	NOT NULL	-1
TimeUnits	INT	NOT NULL	15
CostTypeSeqNo	INT	NOT NULL	0
SupplierId	INT	NOT NULL	-1
CostTypeId	INT	NOT NULL	0

Table: UserConnections - **This table is new**

Fields:

UserId	INT	NOT NULL
ConnectionId	NVARCHAR(50)	NOT NULL

AppCode	NVARCHAR(10)	NOT NULL
UserAgent	NVARCHAR(Max)	NOT NULL
LastResponse	DATETIME	NOT NULL

Table:UserDisbursements- This table is new

Fields:

UserId	INT	NOT NULL
StockCode	VARCHAR(23)	NOT NULL
Rank	INT	NOT NULL
PreLoad	bit	NOT NULL
Quantity	FLOAT	NOT NULL
PurchaseOrderLine	BIT	NOT NULL

Table:UserGroups- This table is updated

Old -GroupMembership

Fields:

UserGroups OLD - GroupMembership		
UserId OLD - UID		
GroupId OLD - GID		
Approval	BIT	NOT NULL

Table:UserLabourCodes- This table is updated

Old -LabourCodeMembership

Fields:

UserId OLD - UID		
StatusId OLD - SID		
RosterClassId	INT	NOT NULL

Table: UserEmailTemplates- This table is updated**Fields:**

Userld OLD - UID	INT	NOT NULL
----------------------------	-----	----------

Table: UserSettings- This table is removed**Fields:****Table: WorkDays - This table is Removed****Fields:**

UID	INT	NOT NULL
DayOfWeek	INT	NOT NULL
DayStart	VARCHAR(8)	NOT NULL
DayEnd	VARCHAR(8)	NOT NULL
MinDailyUnits	FLOAT	NOT NULL
ProdHours	FLOAT	NOT NULL
IsWorkDay	BIT	NOT NULL

SQL Triggers in TimeTrak

New

Renamed

Updated

Removed

Item has been altered - fields may be removed and added

REMOVED - Trigger Name: CreateDefaultGroup

Table: ApplicationUser

Trigger Name: Insert_TaskLog

Table: TT_TASK

FOR INSERT

AS

```

DECLARE      @TaskId As int,
              @UserId As int,
              @AssignedTo As nvarchar(100),
              @Description As varchar(4096),
              @CreatedById As int,
              @CreatedByClient As bit,
              @CreatedByName As nvarchar(100);

-- Get Task Data
SELECT @TaskId = TaskId, @UserId = UserId, @CreatedById = CreatedBy, @CreatedByClient =
ClientCreated FROM Inserted;

IF @TaskId IS NOT NULL
BEGIN
    -- Get who task was created for
    IF @CreatedByClient = 1
        SELECT @CreatedByName = FullName FROM [<DbItem2:Name>].dbo.Contacts WHERE SeqNo =
@CreatedById
    ELSE
        SELECT @CreatedByName = Name FROM Users WHERE Id = @CreatedById;

    -- Get who the task is assigned to
    IF @UserId = 0
        SET @AssignedTo = 'Unallocated'
    ELSE
        SELECT @AssignedTo = Name FROM Users WHERE Id = @UserId;

    -- Build task log description
    If @CreatedByClient = 1
        SET @Description = 'Created in Client Portal by ' + @CreatedByName + ': Assigned to
' + @AssignedTo
    ELSE
        SET @Description = 'Created by ' + @CreatedByName + ': Assigned to ' + @AssignedTo;

    -- Insert new task log
    INSERT INTO TT_TaskLog (TaskId, CreatedBy, CreatedDate, LastModifiedBy, LastModifiedDate,
Description, LogType, ClientVisible, ClientCreated)
    VALUES (@TaskId, @CreatedById, getdate(), @CreatedById, getdate(), @Description, 1, 0,

```

```
@CreatedByClient);
```

```
END
```

Trigger Name: Update_TaskLog

Table: TT_TASK

```
FOR UPDATE
```

```
AS
```

```
DECLARE @Changes As varchar(4096),
        @Name As varchar(200),
        @StatusName as varchar(200),
        @Status As int,
        @OldStatus As int,
        @Priority As int,
        @OldPriority As int,
        @Completed As int,
        @OldCompleted As int,
        @Contact As int,
        @OldContact As int,
        @AssignedTo As int,
        @OldAssignedTo As int,
        @State As int,
        @OldState As int,
        @DueDate As datetime,
        @OldDueDate As datetime,
        @TaskID As int,
        @LastModifiedBy As int,
        @PriorityName As varchar(50);
```

```
SET @Changes = '';
```

```
-- Get data
```

```
SELECT @OldStatus = [Status], @OldPriority = Priority, @OldCompleted = PercentComplete, @OldContact
= Contact, @OldAssignedTo = [UserId], @OldState = [State], @OldDueDate = DueDate FROM Deleted;
SELECT @TaskId = TaskId, @LastModifiedBy = LastModifiedBy, @Status = [Status], @Priority = Priority,
@Completed = PercentComplete, @Contact = Contact, @AssignedTo = [UserId], @State = [State], @DueDate
= DueDate FROM Inserted;
```

```
-- Check if status has changed
```

```
IF (@Status <> @OldStatus)
    BEGIN
        SELECT @StatusName = Name FROM TaskStatuses WHERE ID = @Status;
        SET @Changes = 'Status Changed: ' + @StatusName + '; '
    END
```

```
END
```

```
-- Check if priority has changed
```

```
IF (@Priority <> @OldPriority)
    BEGIN
        SELECT @PriorityName = Name FROM Priorities WHERE Id = @Priority;
        SET @Changes = @Changes + 'Priority Changed: ' + @PriorityName + '; '
    END
```

```
END
```

```
-- Check if percent complete has changed
```

```
IF (@Completed <> @OldCompleted)
    SET @Changes = @Changes + 'Percent Complete Changed: ' + CAST(@Completed As varchar(3))
+ '%; '
```

```

-- Check if contact has changed
IF (@Contact <> @OldContact)
    SET @Changes = @Changes + 'Contact Changed;';

-- Check if state has changed
IF (@State <> @OldState)
    IF (@OldState = 2)
        BEGIN
            SELECT @Name = Name FROM Users WHERE [Id] = @OldAssignedTo;
            IF (@AssignedTo > 0)
                SET @Changes = @Changes + 'Task Accepted; ';
            ELSE
                SET @Changes = @Changes + 'Task Declined; ';
        END

-- Check if assigned To has changed
IF (@AssignedTo <> @OldAssignedTo)
    BEGIN
        SELECT @Name = Name FROM Users WHERE [Id] = @AssignedTo;
        IF (@AssignedTo > 0)
            SET @Changes = @Changes + 'Assigned To: ' + @Name + '; ';
        Else
            SET @Changes = @Changes + 'Assigned To: UNALLOCATED; ';
    END

-- Check due date has changed
IF (@DueDate <> @OldDueDate)
    SET @Changes = @Changes + 'Due Date Changed: ' + CONVERT(varchar(20), @DueDate, 100);

-- Create new change log if required
IF (LEN(@Changes) > 0)
    BEGIN
        INSERT INTO TT_TaskLog (TaskId, CreatedBy, CreatedDate, LastModifiedBy,
LastModifiedDate, Description, LogType, ClientVisible, ClientCreated)
        VALUES (@TaskId, @LastModifiedBy, getdate(), @LastModifiedBy, getdate(), @Changes,
1, 0, 0);
    END
GO

```

SQL Procedures and Views added or edited in TimeTrak

New

Renamed

Updated

Removed

Item has been altered - fields may be removed and added

Remove task processing

```
DELETE FROM SettingValues WHERE FieldId = (SELECT Id FROM SettingFields WHERE Section = 'TimeTrak'
AND Name = 'TaskProcessing');
DELETE FROM SettingFields WHERE Section = 'TimeTrak' AND Name = 'TaskProcessing';
GO
```

View Name: ServiceableUnitsView

SQL Content:

```
SELECT * FROM [<DbItem2:Name>].dbo.SU_MAIN WHERE db_name() = '<DbItem1:Name>'
```

View Name: ClientsView

SQL Content:

```
SELECT * FROM [<DbItem2:Name>].dbo.Dr_Accs WHERE db_name() = '<DbItem1:Name>'
```

View Name: ClientHistoryNotesView

SQL Content:

```
SELECT * FROM [<DbItem2:Name>].dbo.Dr_Cont_Hist WHERE db_name() = '<DbItem1:Name>'
```

View Name: SuppliersView

SQL Content:

```
SELECT * FROM [<DbItem2:Name>].dbo.Cr_Accs WHERE db_name() = '<DbItem1:Name>'
```

View Name: SupplierHistoryNotesView

SQL Content:

```
SELECT * FROM [<DbItem2:Name>].dbo.Cr_Cont_Hist WHERE db_name() = '<DbItem1:Name>'
```

View Name: JobsView

SQL Content:

```
SELECT * FROM [<DbItem2:Name>].dbo.JobCost_Hdr WHERE db_name() = '<DbItem1:Name>'
```

View Name: TaskTotalsView

SQL Content:

```
AS
    WITH
    CTE_TaskTotals As
    (
        SELECT T.TaskId, T.ParentId, T.JobNo, T.Status, T.GroupId, T.UserId, T.TotalSubTask,
```



```

T.IsActive, T.TimeRequired,
        COALESCE((SELECT SUM(Duration) FROM TimeEntry E JOIN Status S ON S.Sid = E.Sid
AND S.LineStatus <> 2 WHERE E.IsActive = 1 AND E.TaskId = T.TaskId), 0) AS ActualTime,
        COALESCE((SELECT SUM(Duration) FROM Appointments WHERE LinkId = T.TaskId AND
LinkType = 2 AND IsActive = 1 AND AppointmentType = 0), 0) AS ScheduledTime,
        COALESCE((SELECT COUNT(Id) FROM Appointments WHERE LinkId = T.TaskId AND
LinkType = 2 AND IsActive = 1 AND AppointmentType = 0), 0) AS AppointmentCount,
        COALESCE((SELECT COUNT(TaskId) FROM TT_Task WHERE IsActive = 1 AND ParentId
= T.TaskId), 0) As SubTasks
        FROM TT_Task T
        WHERE IsActive = 1
    ),
    CTE_Recursive As
    (
        SELECT T.TaskId, T.TaskId AS RootId,
            CASE T.TotalSubTask WHEN 1 THEN 0 ELSE T.TimeRequired END As EstimatedTime,
            CASE T.TotalSubTask WHEN 1 THEN 0 ELSE T.ActualTime END As ActualTime,
            CASE T.TotalSubTask WHEN 1 THEN 0 ELSE T.ScheduledTime END As
ScheduledTime,
            CASE T.TotalSubTask WHEN 1 THEN 0 ELSE T.AppointmentCount END As
AppointmentCount,
            T.SubTasks
        FROM CTE_TaskTotals T

        UNION ALL

        SELECT T.TaskId, R.RootId,
            CASE T.TotalSubTask WHEN 1 THEN 0 ELSE T.TimeRequired END As
EstimatedTime,
            CASE T.TotalSubTask WHEN 1 THEN 0 ELSE T.ActualTime END As
ActualTime,
            CASE T.TotalSubTask WHEN 1 THEN 0 ELSE T.ScheduledTime END As
ScheduledTime,
            CASE T.TotalSubTask WHEN 1 THEN 0 ELSE T.AppointmentCount END
As AppointmentCount,
            T.SubTasks
        FROM CTE_TaskTotals T INNER JOIN CTE_Recursive R ON T.ParentId = R.TaskId AND
T.ParentId > 0
    )
    SELECT RootId AS TaskId,
        SUM(C.EstimatedTime) As EstimatedTime,
        SUM(C.ActualTime) As ActualTime,
        SUM(C.ScheduledTime) As ScheduledTime,
        SUM(C.AppointmentCount) As AppointmentCount,
        SUM(C.SubTasks) AS SubTasks
    FROM CTE_Recursive C
    GROUP BY RootId;

```

Stored Procedure Name: AddStatus - Updated

SQL Content:

```

( @Status As varchar(50),

                                @Export As bit,
                                @NonProductive As bit,
                                @Colour As varchar(7),
                                @NoValue As bit,
                                @PayrollCode As varchar(20),
                                @CostCentre As varchar(20),
                                @Multiplier As float,

```

```

        @WageCode As int,
        @ShortcutKey As varchar(1),
        @BranchNo As int,
        @LineStatus As int,
        @DisbLineStatus As int,
        @IncludeInProdCalc As bit,
        @IncludeInBudget As bit,
        @SID As int OUTPUT
    )
AS
BEGIN TRANSACTION
INSERT INTO Status (Status, Export, NonProductive, Colour, NoValue, PayrollCode, CostCentre,
Multiplier, WageCode, ShortcutKey, BranchNo, LineStatus, DisbLineStatus, IncludeInProdCalc,
IncludeInBudget)
VALUES(@Status, @Export, @NonProductive, @Colour, @NoValue, @PayrollCode, @CostCentre,
@Multiplier, @WageCode, @ShortcutKey, @BranchNo, @LineStatus, @DisbLineStatus,
@IncludeInProdCalc, @IncludeInBudget);
COMMIT TRANSACTION
SELECT @SID = MAX(SID) FROM Status;

```

Stored Procedure Name: UpdateStatus - Updated

SQL Content:

```

( @SID As int,
    @Status As varchar(50),
    @Export As bit,
    @NonProductive As bit,
    @Colour As varchar(7),
    @InUse As bit,
    @NoValue As bit,
    @PayrollCode As
    varchar(20),
    @CostCentre As
    varchar(20),
    @Multiplier As float,
    @WageCode As int,
    @ShortcutKey As
    varchar(1),
    @BranchNo As int,
    @LineStatus As int,
    @DisbLineStatus As int,
    @IncludeInProdCalc As bit,
    @IncludeInBudget AS bit
)
AS
BEGIN TRANSACTION
UPDATE Status SET Status = @Status, Export = @Export, NonProductive = @NonProductive, InUse = @InUse,
Colour = @Colour, NoValue = @NoValue, PayrollCode = @PayrollCode, CostCentre = @CostCentre,
Multiplier = @Multiplier, WageCode = @WageCode, ShortcutKey = @ShortcutKey, BranchNo = @BranchNo,
LineStatus = @LineStatus, DisbLineStatus = @DisbLineStatus, IncludeInProdCalc =
@IncludeInProdCalc, IncludeInBudget = @IncludeInBudget
WHERE SID = @SID;
COMMIT TRANSACTION

```

Stored Procedure Name: AddUser - Updated

SQL Content:

```

( @LoginID AS varchar(20),
  @Username AS varchar(100),
  @ShortName AS nvarchar(8),

```

```

@Initials As nvarchar(3),
@Email AS varchar(100),
@StaffID AS int,
@Password AS varchar(100),
@Image As varbinary(MAX),
@LatestCost AS float,
@GroupAdministrator AS bit,
@GlobalAdministrator AS bit,
@OverwriteSetup AS bit,
@Style AS varchar(50),
@CanAddTime AS bit,
@LastChecked AS DateTime,
@ShowRate AS bit,
@DaysOnCalendar AS int,
@OutlookProfile AS varchar(100),
@TimeFormat AS int,
@loginMessage AS varchar(4096),
@TaskTrakUser AS bit,
@StockLocation AS int,
@PayrollCode AS varchar(50),
@PasswordExpiryDate AS datetime,
@UserType AS int,
@TaskView AS int,
@DateView AS int,
@ExpandLevel AS int,
@RestrictedUser AS bit,
@BasePayRateCode AS varchar(20),
@BranchNo AS int,
@TaskEmail AS bit,
@TaskTrakView As int,
@LocationName As varchar(50),
@Latitude As float,
@Longitude As float,
@ZoomLevel AS int,
@ClosedTasksVisible As bit,
@RefreshMins AS int,
@ShowAllSubTasks AS bit,
@NewTaskAssignment AS int,
@TaskAcceptedEmail As bit,
@WageCostType As int,
@ShowAcceptanceTab As bit,
@CalendarType As int,
@CalendarClick As int,
@PreLoadTaskTitle As bit,
@DefaultTaskSortField As int,
@DefaultTaskSortOrder As int,
@ProfileId As int,
@MobileUser As bit,
@CreatedById As int,
@CreatedDate As datetime,
@StartDate As datetime,
@UID AS int OUTPUT
)
AS
BEGIN TRANSACTION
INSERT INTO ApplicationUser (LoginID, Username, ShortName, Initials, Email, StaffID, Password,
Image, LatestCost, GroupAdministrator, GlobalAdministrator, OverwriteSetup, Style, CanAddTime,
LastChecked, ShowRate, DaysOnCalendar, OutlookProfile, TimeFormat, LoginMessage, TaskTrakUser,
StockLocation, PayrollCode, PasswordExpiryDate, UserType, TaskView, DateView, ExpandLevel,
RestrictedUser, BasePayRateCode, BranchNo, TaskEmail, TaskTrakView, LocationName, Latitude,
Longitude, ZoomLevel, ClosedTasksVisible, RefreshMins, ShowAllSubTasks, NewTaskAssignment,

```

```

TaskAcceptedEmail, WageCostType, ShowAcceptanceTab, CalendarType, CalendarClick,
PreLoadTaskTitle, DefaultTaskSortField, DefaultTaskSortOrder, ProfileId, MobileUser, CreatedById,
CreatedDate, StartDate)
VALUES(@LoginID, @Username, @ShortName, @Initials, @Email, @StaffID, @Password, @Image,
@LatestCost, @GroupAdministrator, @GlobalAdministrator, @OverwriteSetup, @Style, @CanAddTime,
@LastChecked, @ShowRate, @DaysOnCalendar, @OutlookProfile, @TimeFormat, @LoginMessage,
@TaskTrakUser, @StockLocation, @PayrollCode, @PasswordExpiryDate, @UserType, @TaskView,
@DateView, @ExpandLevel, @RestrictedUser, @BasePayRateCode, @BranchNo, @TaskEmail, @TaskTrakView,
@LocationName, @Latitude, @Longitude, @ZoomLevel, @ClosedTasksVisible, @RefreshMins,
@ShowAllSubTasks, @NewTaskAssignment, @TaskAcceptedEmail, @WageCostType, @ShowAcceptanceTab,
@CalendarType, @CalendarClick, @PreLoadTaskTitle, @DefaultTaskSortField, @DefaultTaskSortOrder,
@ProfileId, @MobileUser, @CreatedById, @CreatedDate, @StartDate);
COMMIT TRANSACTION
SELECT @UID = MAX(UID) FROM ApplicationUser;

GO

```

Stored Procedure Name: UpdateUser - Updated

SQL Content:

```

( @UID AS int,
  @LoginID AS varchar(20),
  @Username AS varchar(100),
  @ShortName AS nvarchar(8),
  @Initials AS nvarchar(3),
  @Email AS varchar(100),
  @StaffID AS int,
  @Password AS varchar(100),
  @Image AS varbinary(MAX),
  @LatestCost AS float,
  @GroupAdministrator AS bit,
  @GlobalAdministrator AS bit,
  @OverwriteSetup AS bit,
  @Style AS varchar(50),
  @CanAddTime AS bit,
  @LastChecked AS DateTime,
  @ShowRate AS bit,
  @DaysOnCalendar AS int,
  @OutlookProfile AS varchar(100),
  @TimeFormat AS int,
  @LoginMessage AS varchar(4096),
  @TaskTrakUser AS bit,
  @StockLocation AS int,
  @PayrollCode AS varchar(50),
  @PasswordExpiryDate AS datetime,
  @UserType AS int,
  @TaskView AS int,
  @DateView AS int,
  @ExpandLevel AS int,
  @RestrictedUser AS bit,
  @BasePayRateCode AS varchar(20),
  @BranchNo AS int,
  @TaskEmail AS bit,
  @TaskTrakView AS int,
  @LocationName AS varchar(50),
  @Latitude AS float,
  @Longitude AS float,
  @ZoomLevel AS int,

```

```

@ClosedTasksVisible As bit,
@RefreshMins As int,
@ShowAllSubTasks AS bit,
@NewTaskAssignment AS int,
@TaskAcceptedEmail AS bit,
@WageCostType As int,
@ShowAcceptanceTab As int,
@CalendarType As int,
@CalendarClick As int,
@PreLoadTaskTitle As bit,
@DefaultTaskSortField As int,
@DefaultTaskSortOrder As int,
@ProfileId As int,
@MobileUser As bit,
@CreatedById As int,
@CreatedDate As datetime,
@StartDate As datetime
)
AS
BEGIN TRANSACTION
UPDATE ApplicationUser SET LoginID = @LoginID, Username = @Username, ShortName = @ShortName,
Initials = @Initials, Email = @Email, StaffID = @StaffID, Password = @Password, Image = @Image,
LatestCost = @LatestCost, GroupAdministrator = @GroupAdministrator, GlobalAdministrator =
@GlobalAdministrator, OverwriteSetup = @OverwriteSetup, Style = @Style, CanAddTime = @CanAddTime,
LastChecked = @LastChecked, ShowRate = @ShowRate, DaysOnCalendar = @DaysOnCalendar, OutlookProfile
= @OutlookProfile, TimeFormat = @TimeFormat, LoginMessage = @LoginMessage, TaskTrakUser =
@TaskTrakUser, StockLocation = @StockLocation, PayrollCode = @PayrollCode, PasswordExpiryDate =
@PasswordExpiryDate, UserType = @UserType, TaskView = @TaskView, DateView = @DateView, ExpandLevel
= @ExpandLevel, RestrictedUser = @RestrictedUser, BasePayRateCode = @BasePayRateCode, BranchNo =
@BranchNo, TaskEmail = @TaskEmail, TaskTrakView = @TaskTrakView, LocationName = @LocationName,
Latitude = @Latitude, Longitude = @Longitude, ZoomLevel = @ZoomLevel, ClosedTasksVisible =
@ClosedTasksVisible, RefreshMins = @RefreshMins, ShowAllSubTasks = @ShowAllSubTasks,
NewTaskAssignment = @NewTaskAssignment, TaskAcceptedEmail = @TaskAcceptedEmail, WageCostType =
@WageCostType, ShowAcceptanceTab = @ShowAcceptanceTab, CalendarType = @CalendarType, CalendarClick
= @CalendarClick, PreLoadTaskTitle = @PreLoadTaskTitle, DefaultTaskSortField =
@DefaultTaskSortField, DefaultTaskSortOrder = @DefaultTaskSortOrder, ProfileId = @ProfileId,
MobileUser = @MobileUser, CreatedById = @CreatedById, CreatedDate = @CreatedDate, StartDate =
@StartDate
WHERE UID = @UID;
COMMIT TRANSACTION

GO

```

Stored Procedure Name: AddTimeEntry - Updated

SQL Content:

```

( @UserId As int,

nvarchar(40),

varchar(23),

@JobNo As int,
@EntryDate As datetime,
@StartTime As datetime,
@EndTime As datetime,
@StartTimeUtc As datetime,
@EndTimeUtc As datetime,
@TimeZoneId As

@Sid As int,
@TimeAnalysisId As int,
@LabourCode As

@Rate As float,

```

```

varchar(255),

varchar(4096),

datetime,

nvarchar(50),

@BillingQuery As

@OrderNo As varchar(255),
@Narration As

@CostGroup As int,
@CostType As int,
@TaskId As int,
@State As int,
@Locked As bit,
@LatestCost As float,
@Checked As bit,
@CurrencyRate As float,
@Verified1 As bit,
@Verified2 As bit,
@Approval As int,
@ApprovalId As int,
@Travel As bit,
@CreatedById As int,
@CreatedDate As datetime,
@LastModifiedById As int,
@LastModifiedDate As

@SplitLineId As int,
@RelatedGuid As

@Notes As nvarchar(MAX),
@Eid As int OUTPUT
)

AS
BEGIN TRANSACTION
INSERT INTO TimeEntry (UserId, JobNo, EntryDate, StartTime, EndTime, StartTimeUtc, EndTimeUtc,
TimeZoneId, SID, TimeAnalysisId, LabourCode, Rate, BillingQuery, OrderNo, Narration, CostGroup,
CostType, TaskID, State, Locked, LatestCost, Checked, CurrencyRate, Verified1, Verified2, Approval,
ApprovalId, Travel, CreatedById, CreatedDate, LastModifiedById, LastModifiedDate, SplitLineId,
RelatedGuid, Notes)
VALUES(@UserId, @JobNo, @EntryDate, @StartTime, @EndTime, @StartTimeUtc, @EndTimeUtc, @TimeZoneId,
@SID, @TimeAnalysisId, @LabourCode, @Rate, @BillingQuery, @OrderNo, @Narration, @CostGroup,
@CostType, @TaskID, @State, @Locked, @LatestCost, @Checked, @CurrencyRate, @Verified1, @Verified2,
@Approval, @ApprovalId, @Travel, @CreatedById, @CreatedDate, @LastModifiedById,
@LastModifiedDate, @SplitLineId, @RelatedGuid, @Notes);
COMMIT TRANSACTION
SELECT @EID = MAX(EID) FROM TimeEntry WHERE UserId = @UserId AND EntryDate = @EntryDate AND StartTime
= @StartTime AND EndTime = @EndTime And IsActive = 1
GO

```

Stored Procedure Name: UpdateTimeEntry - Updated

SQL Content:

```

(
    @Eid As int,

@UserId As int,
@JobNo As int,
@EntryDate As

DateTime,

@StartTime As

DateTime,

@EndTime As

DateTime,

@StartTimeUtc As

datetime,

```

```

datetime,
nvarchar(40),
int,
varchar(23),
varchar(255),
varchar(255),
varchar(4096),

float,

float,

datetime,
int,
datetime,

nvarchar(50),
nvarchar(MAX)

@EndTimeUtc As
@TimeZoneId As
@Sid As int,
@TimeAnalysisId As
@LabourCode As
@Rate As float,
@BillingQuery As
@OrderNo As
@Narration As
@CostGroup As int,
@CostType As int,
@TaskId As int,
@State As int,
@Locked As bit,
@LatestCost As
@Checked As bit,
@CurrencyRate As
@Verified1 As bit,
@Verified2 As bit,
@Approval As int,
@ApprovalId As int,
@Travel As bit,
@CreatedById As int,
@CreatedDate As
@LastModifiedById As
@LastModifiedDate As
@SplitLineId As int,
@RelatedGuid As
@Notes As
)
AS
BEGIN TRANSACTION
UPDATE TimeEntry SET UserId = @UserId, JobNo = @JobNo, EntryDate = @EntryDate, StartTime =
@StartTime, EndTime = @EndTime, StartTimeUtc = @StartTimeUtc, EndTimeUtc = @EndTimeUtc, TimeZoneId
= @TimeZoneId, Sid = @Sid, TimeAnalysisId = @TimeAnalysisId, LabourCode = @LabourCode, Rate = @Rate,
BillingQuery = @BillingQuery, OrderNo = @OrderNo, Narration = @Narration, CostGroup = @CostGroup,
CostType = @CostType, TaskId = @TaskId, State = @State, Locked = @Locked, LatestCost = @LatestCost,
Checked = @Checked, CurrencyRate = @CurrencyRate, Verified1 = @Verified1, Verified2 = @Verified2,
Approval = @Approval, ApprovalId = @ApprovalId, Travel = @Travel, CreatedById = @CreatedById,
CreatedDate = @CreatedDate, LastModifiedById = @LastModifiedById, LastModifiedDate =
@LastModifiedDate, SplitLineId = @SplitLineId, RelatedGuid = @RelatedGuid, Notes = @Notes
WHERE Eid = @Eid;
COMMIT TRANSACTION
GO

```

Stored Procedures : For Task Types, Categories, and Task Logs - [Removed](#)

SQL Content:

```

/*# Message: Drop Task Type Question Stored Procedures #*/
DROP PROCEDURE [dbo].[TT_AddQuestion];
DROP PROCEDURE [dbo].[TT_UpdateQuestion];
DROP PROCEDURE [dbo].[TT_DeleteQuestion];

/*# Message: Drop Task Type Stored Procedures #*/
DROP PROCEDURE [dbo].[TT_AddJobType];
DROP PROCEDURE [dbo].[TT_UpdateJobType];
DROP PROCEDURE [dbo].[TT_DeleteJobType];

/*# Message: Drop Task Categories Stored Procedures #*/
DROP PROCEDURE [dbo].[TT_AddCategory];
DROP PROCEDURE [dbo].[TT_UpdateCategory];
DROP PROCEDURE [dbo].[TT_DeleteCategory];

/*# Message: Drop Task Type Stored Procedures #*/
DROP PROCEDURE [dbo].[TT_AddStatus];
DROP PROCEDURE [dbo].[TT_UpdateStatus];
DROP PROCEDURE [dbo].[TT_DeleteStatus];

/*# Message: Drop Task Log Triggers #*/
DROP PROCEDURE [dbo].[TT_AddStatus];
DROP PROCEDURE [dbo].[TT_UpdateStatus];
DROP PROCEDURE [dbo].[TT_DeleteStatus];

/*# Message: Remove unused stored procedures #*/
drop procedure [dbo].[TC_AddCustomerGroup];
drop procedure [dbo].[TC_UpdateCustomerGroup];
drop procedure [dbo].[TC_DeleteCustomerGroup];
drop procedure [dbo].[TC_AddWizardPage];
drop procedure [dbo].[TC_UpdateWizardPage];
drop procedure [dbo].[TC_DeletewizardPage];
drop procedure [dbo].[TC_AddTaskField];
drop procedure [dbo].[TC_UpdateTaskField];
drop procedure [dbo].[TC_DeleteTaskField];

```

Stored Procedure Name: TT_AddTask - Updated**SQL Content:**

```

(    @JobNo As int,
                                     @Title As varchar(100),
                                     @Description As
                                     @Contact as int,
                                     @UserId As int,
                                     @GroupId As int,
                                     @JobType As int,
                                     @Priority As int,
                                     @JobCategory as int,
                                     @Status As int,
                                     @PercentComplete as int,
                                     @TimeRequired as float,
                                     @Notes As varchar(4096),
                                     @CreatedBy As int,

```



```

        @AssignedBy As int,
        @StartTime As datetime,
        @DueDate as datetime,
        @DateEntered As datetime,
        @DateClosed as datetime,
        @ParentID as int,
        @State as int,
        @RID AS int,
        @RecurrenceEnabled AS bit,
        @SyncTask as bit,
        @TotalSubTask AS bit,
        @AllowTimeEntries As bit,
        @SameDayCompletion As bit,
        @IgnoreBreaks As bit,
        @OrderNo As varchar(255),
        @Urgency As int,
        @ClientCreated As bit,
        @SID As int,
        @GlobalTask As bit,
        @TaskExpires As bit,
        @ExpiryDate As datetime,
        @ResponseLevel As int,
        @Locked As bit,
        @LockedById As int,
        @LockedDate As datetime,
        @BillingQuery As

nvarchar(255),

        @ClientVisible As bit,
        @RelatedGuid As

nvarchar(50),

        @TaskId as int OUTPUT
    )

AS

DECLARE @found as int;
SELECT @found = count(taskID) FROM TT_Task WHERE JobNo = @JobNo AND Title = @Title AND Description
= @Description And Contact = @Contact And UserId = @UserId And GroupId = @GroupId And JobType =
@JobType And JobCategory = @JobCategory And Status = @Status And StartTime = @StartTime And ParentID
= @ParentID And IsActive = 1
IF (@found = 0)
BEGIN
    BEGIN TRANSACTION
        INSERT INTO TT_Task (JobNo, Title, Description, Contact, UserId, GroupId, JobType, Priority,
JobCategory, Status, PercentComplete, TimeRequired, Notes, CreatedBy, AssignedBy, LastModifiedBy,
LastModifiedByClient, LastModifiedDate, StartTime, DueDate, DateEntered, DateClosed, ParentID,
State, RID, RecurrenceEnabled, SyncTask, TotalSubTask, AllowTimeEntries, SameDayCompletion,
IgnoreBreaks, OrderNo, Urgency, ClientCreated, SID, GlobalTask, TaskExpires, ExpiryDate,
ResponseLevel, Locked, LockedById, LockedDate, BillingQuery, ClientVisible, RelatedGuid)
        VALUES(@JobNo, @Title, @Description, @Contact, @UserId, @GroupId, @JobType, @Priority,
@JobCategory, @Status, @PercentComplete, @TimeRequired, @Notes, @CreatedBy, @AssignedBy,
@CreatedBy, @ClientCreated, @DateEntered, @StartTime, @DueDate, @DateEntered, @DateClosed,
@ParentID, @State, @RID, @RecurrenceEnabled, @SyncTask, @TotalSubTask, @AllowTimeEntries,
@SameDayCompletion, @IgnoreBreaks, @OrderNo, @Urgency, @ClientCreated, @SID, @GlobalTask,
@TaskExpires, @ExpiryDate, @ResponseLevel, @Locked, @LockedById, @LockedDate, @BillingQuery,
@ClientVisible, @RelatedGuid);
    COMMIT TRANSACTION
END;
SELECT @TaskID = TaskID FROM TT_Task WHERE JobNo = @JobNo AND Title = @Title AND Description =
@Description And Contact = @Contact And UserId = @UserId And GroupId = @GroupId And JobType = @JobType
And JobCategory = @JobCategory And Status = @Status And StartTime = @StartTime And ParentID =
@ParentID And IsActive = 1

```

GO

Stored Procedure Name: TT_UpdateTask - **Updated**
SQL Content:

```
( @TaskID AS int,
varchar(100),
varchar(4096),

int,
float,
varchar(4096),
int,
@LastModifiedByClient As int,
datetime,
datetime,
datetime,
datetime,
datetime,
AS bit,
bit,
bit,
As bit,
bit,
varchar(255),
bit,

@JobNo AS int,
@Title AS
@Description AS
@Contact AS int,
@UserId AS int,
@GroupId AS int,
@JobType AS int,
@Priority AS int,
@JobCategory AS int,
@Status AS int,
@PercentComplete AS
@TimeRequired AS
@Notes AS
@LastModifiedBy AS
@LastModifiedDate As
@AssignedBy AS int,
@StartTime AS
@DueDate AS
@DateEntered AS
@DateClosed AS
@ParentID AS int,
@State AS int,
@RID AS int,
@RecurrenceEnabled
@SyncTask AS bit,
@TotalSubTask AS
@AllowTimeEntries as
@SameDayCompletion
@IgnoreBreaks As
@OrderNo As
@Urgency As int,
@ClientCreated As
@SID As int,
```

```

datetime,
int,
datetime,
nvarchar(255),
bit,
nvarchar(50)
)
AS
BEGIN TRANSACTION
UPDATE TT_Task SET JobNo = @JobNo, Title = @Title, Description = @Description, Contact = @Contact,
UserId = @UserId, GroupId = @GroupId, JobType = @JobType, Priority = @Priority, JobCategory =
@JobCategory, Status = @Status, PercentComplete = @PercentComplete, TimeRequired = @TimeRequired,
Notes = @Notes, LastModifiedBy = @LastModifiedBy, LastModifiedByClient = @LastModifiedByClient,
LastModifiedDate = @LastModifiedDate, AssignedBy = @AssignedBy, StartTime = @StartTime, DueDate
= @DueDate, DateEntered = @DateEntered, DateClosed = @DateClosed, ParentID = @ParentID, State =
@State, RID = @RID, RecurrenceEnabled = @RecurrenceEnabled, SyncTask = @SyncTask, TotalSubTask =
@TotalSubTask, AllowTimeEntries = @AllowTimeEntries, SameDayCompletion = @SameDayCompletion,
IgnoreBreaks = @IgnoreBreaks, OrderNo = @OrderNo, Urgency = @Urgency, ClientCreated =
@ClientCreated, SID = @SID, GlobalTask = @GlobalTask, TaskExpires = @TaskExpires, ExpiryDate =
@ExpiryDate, ResponseLevel = @ResponseLevel, Locked = @Locked, LockedById = @LockedById, LockedDate
= @LockedDate, BillingQuery = @BillingQuery, ClientVisible = @ClientVisible, RelatedGuid =
@RelatedGuid
WHERE TaskID = @TaskID;
COMMIT TRANSACTION
GO

```