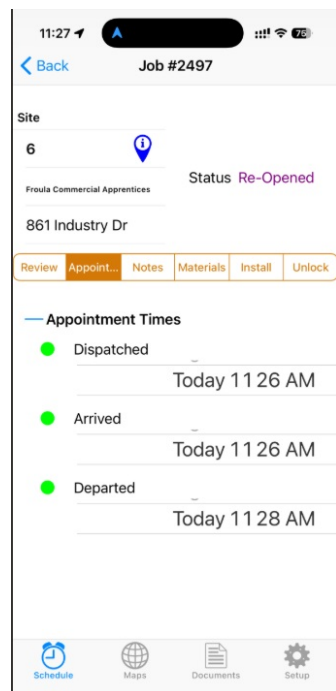


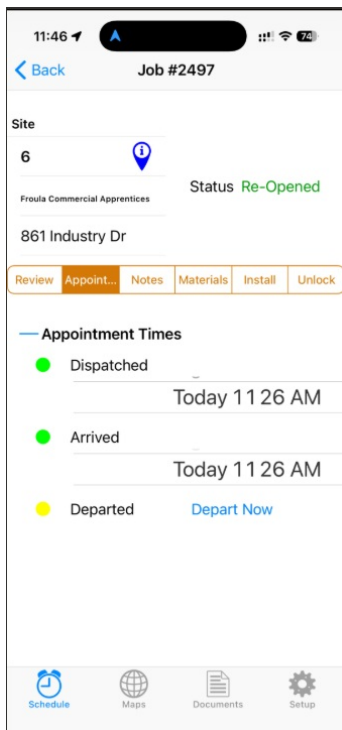
Techs unable to depart/save notes in Legacy FSU after updating to 6.2.0.23 rev3

09/17/2026 9:10 am EDT

170056, 169067, 168206, 167845

Technicians are no longer able to "depart" from jobs on FSU. It works fine for Tickets, and works from Sedona X or the Sedona Client side. There is no error , the ticket will not capture the time.





The following two scripts created by Scott P need to be ran on their company DB associated with Legacy FSU.

FSU_Job_Appointment_Upd

```

/***** Object: StoredProcedure [dbo].[FSU_Job_Appointment_Upd]  Script Date: 9/2/2026 10:00:43 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROC [dbo].[FSU_Job_Appointment_Upd]
    @dispatch_id int,
    @job_id int,
    @service_tech_id int,
    @schedule_time datetime,
    @dispatch_time datetime,
    @arrival_time datetime,
    @departure_time datetime,
    @estimated_length int,
    @taskcode varchar(25),
    @labortaskcode varchar(25),
    @timesheet_id int,
    @completed char(1),
    @usercode varchar(30),
    @edit_timestamp datetime

AS
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

DECLARE @task_id int

```

```
DECLARE @task_id int,  
        @labor_task_id int,
```

```
--6/19/07 JMS Added for Auto Timesheet version 5.0.12
```

```
@job_code varchar(25),  
@auto_ts char(1),  
@ts_from char(1),  
@job_task_id int,  
@installer_code varchar(25),  
@work_date datetime,  
@units money,  
@rate money,  
@prevwage money,  
@amount money,  
@posted_to_gl char(1), --6/19/07 JMS not in use. Set to N  
@entered datetime,  
@rate_type char(1)
```

```
--IF EXISTS (SELECT Dispatch_Id FROM OE_Job_Dispatch WHERE Service_Tech_Id = @service_tech_id AND Schedule_Time  
= @schedule_time AND Dispatch_Id <> @dispatch_id)
```

```
--BEGIN
```

```
-- RETURN
```

```
--END
```

```
SELECT @job_Code = Job_Code FROM OE_Job WHERE Job_Id = @job_id
```

```
EXEC Get_Labor_Task_id @labortaskcode, @labor_task_id OUTPUT
```

```
EXEC Get_Task_id @taskcode, @task_id OUTPUT
```

```
IF @timesheet_id < 2 -- No timesheet link was passed so let's be sure there isn't one so we don't create duplicates..
```

```
BEGIN
```

```
SELECT @timesheet_id = ISNULL((SELECT Timesheet_Id FROM OE_Job_Dispatch WHERE Dispatch_Id = @dispatch_id), 1) -
```

```
- Preventing duplicated entries
```

```
END
```

```
UPDATE OE_Job_Dispatch
```

```
SET Service_Tech_Id = @service_tech_id,
```

```
    Schedule_Time = @schedule_time,
```

```
    Dispatch_Time = @dispatch_time,
```

```
    Arrival_Time = @arrival_time,
```

```
    Departure_Time = @departure_time,
```

```
    Completed = @completed,
```

```
    Estimated_Length = @estimated_length,
```

```
    Task_Id = @task_id,
```

```
    Labor_Task_id = @labor_task_id,
```

```
    UserCode = @usercode,
```

```
    Edit_Timestamp = @edit_timestamp
```

```
WHERE Dispatch_Id = @dispatch_id
```

```
IF @service_tech_id > 1 AND @dispatch_time > 0
```

```
    BEGIN
```

```
        UPDATE OE_Job_Task
```

```
        SET Last_Service_Tech_Id = @service_tech_id, Last_Dispatch_Date = @dispatch_time
```

```
        WHERE Job_Id = @job_id AND Job_Task_Id = @task_id
```

```
    END
```

```
SELECT @auto_ts = Timesheet, @ts_from = Timesheet_From, @posted_to_gl = 'N' FROM OE_Install_Company ic INNER JOI
```

```

N SV_Service_Tech tech ON tech.Install_Company_Id = ic.Install_Company_Id WHERE tech.Service_Tech_id = @service_tec
h_id
IF @auto_ts = 'A' AND @departure_time <> {d '1899-12-30'}
BEGIN
    SELECT @job_task_id = Job_Task_Id FROM OE_Job_Task jt INNER JOIN OE_Task task ON task.Task_Id = jt.Task_id WHERE t
ask.Task_Code = @taskcode AND jt.Job_id = @job_id
    SELECT @installer_code = emp.Employee_Code, @rate = tech.RegularPayRate FROM SV_Service_Tech tech INNER JOIN
SY_Employee emp ON emp.Employee_Id = tech.Employee_Id WHERE Service_Tech_Id = @service_tech_id
    SELECT @prevwage = Prevailing_Wage FROM OE_Job WHERE Job_Id = @job_id
    SET @rate_type = 'R'
    IF @prevwage > 0
    BEGIN
        SET @rate = @prevwage
        SET @rate_type = 'P'
    END
    SELECT @work_date = CAST(CONVERT ( char(10), @arrival_time, 102 ) as Datetime)
    SELECT @entered = GETDATE()
    IF @ts_from = 'A'
    BEGIN
        SET @units = CONVERT(money,DATEDIFF ( n , @arrival_time, @departure_time))/60
--    SELECT @units = DATEDIFF ( hour , @arrivaltime , @departure) + (((DATEDIFF ( n , @arrivaltime , @departure)) % 60
) * .01)
    END
    ELSE
    BEGIN
        SET @units = CONVERT(money,DATEDIFF ( n , @dispatch_time, @departure_time))/60
--    SELECT @units = DATEDIFF ( hour , @dispatchtime , @departure) + (((DATEDIFF ( n , @dispatchtime , @departure))
% 60) * .01)
    END
    SELECT @amount = @rate * @units

    IF @timesheet_id > 1
    BEGIN
        EXEC Job_Timesheet_UPD @timesheet_id, @installer_code, @work_date, @labortaskcode, @units, @rate,
        @amount, 'Dispatch Process Auto Created Based On Install Company', @posted_to_gl, @rate_type, @usercode,
        @entered
    END
    ELSE
    BEGIN
        EXEC Job_Timesheet_ADD @job_code, @job_task_id, @installer_code, @work_date, @labortaskcode, @units, @rate,
        @amount, 'Dispatch Process Auto Created Based On Install Company', @posted_to_gl, @usercode,
        @entered, @rate_type, 0, @timesheet_id OUTPUT
        UPDATE OE_Job_Dispatch SET Timesheet_Id = @timesheet_id WHERE Dispatch_Id = @dispatch_id
    END
END
RETURN

```

FSUV3_Job_Appointment_Upd

```

/***** Object: StoredProcedure [dbo].[FSUV3_Job_Appointment_Upd]  Script Date: 9/2/2026 9:41:03 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```
ALTER PROC[dbo].[FSUV3_Job_Appointment_Upd]
```

```
@dispatch_id int,  
@job_id int,  
    @service_tech_id int,  
    @schedule_time datetime,  
    @dispatch_time datetime,  
    @arrival_time datetime,  
    @departure_time datetime,  
    @estimated_length int,  
    @taskcode varchar(25),  
@labortaskcode varchar(25),  
    @timesheet_id int,  
    @completed char(1),  
    @usercode varchar(30),  
    @edit_timestamp datetime
```

```
AS
```

```
-- SET NOCOUNT ON added to prevent extra result sets from  
-- interfering with SELECT statements.
```

```
SET NOCOUNT ON;
```

```
DECLARE @task_id int,
```

```
    @labor_task_id int,
```

```
--6/19/07 JMS Added for Auto Timesheet version 5.0.12
```

```
    @job_code varchar(25),
```

```
    @auto_ts char(1),
```

```
    @ts_from char(1),
```

```
    @job_task_id int,
```

```
    @installer_code varchar(25),
```

```
    @work_date datetime,
```

```
    @units money,
```

```
    @rate money,
```

```
    @prevwage money,
```

```
    @amount money,
```

```
    @posted_to_gl char(1), --6/19/07 JMS not in use. Set to N
```

```
    @entered datetime,
```

```
    @rate_type char(1)
```

```
--IF EXISTS (SELECT Dispatch_Id FROM OE_Job_Dispatch WHERE Service_Tech_Id = @service_tech_id AND Schedule_Time  
= @schedule_time AND Dispatch_Id <> @dispatch_id)
```

```
--BEGIN
```

```
-- RETURN
```

```
--END
```

```
SELECT @job_Code = Job_Code FROM OE_Job WHERE Job_Id = @job_id
```

```
EXEC Get_Labor_Task_id @labortaskcode, @labor_task_id OUTPUT
```

```
EXEC Get_Task_id @taskcode, @task_id OUTPUT
```

```
IF @timesheet_id < 2 -- No timesheet link was passed so let's be sure there isn't one so we don't create duplicates..
```

```
BEGIN
```

```
    SELECT @timesheet_id = ISNULL((SELECT Timesheet_Id FROM OE_Job_Dispatch WHERE Dispatch_Id = @dispatch_id), 1) -
```

```
- Preventing duplicated entries
```

```
END
```

```
UPDATE OE_Job_Dispatch
```

```
SET Service_Tech_Id = @service_tech_id,
```

```
    Schedule_Time = @schedule_time,
```

```
    Dispatch_Time = @dispatch_time,
```

```
    Arrival_Time = @arrival_time,
```

```
    Departure_Time = @departure_time,
```

```
    Completed = @completed,
```

```
    Estimated_Length = @estimated_length
```

```

        Estimated_Length = @estimated_length,
        Task_Id = @task_Id,
        Labor_Task_id = @labor_task_id,
        UserCode = @usercode,
        Edit_Timestamp = @edit_timestamp
WHERE Dispatch_Id = @dispatch_id
SELECT @auto_ts = Timesheet, @ts_from = Timesheet_From, @posted_to_gl = 'N' FROM OE_Install_Company ic INNER JOIN
SV_Service_Tech tech ON tech.Install_Company_Id = ic.Install_Company_Id WHERE tech.Service_Tech_id = @service_tec
h_id
IF @auto_ts = 'A' AND @departure_time <> {d '1899-12-30'}
BEGIN
    SELECT @job_task_id = Job_Task_Id FROM OE_Job_Task jt INNER JOIN OE_Task task ON task.Task_Id = jt.Task_id WHERE t
ask.Task_Code = @taskcode AND jt.Job_id = @job_id
    SELECT @installer_code = emp.Employee_Code, @rate = tech.RegularPayRate FROM SV_Service_Tech tech INNER JOIN
SY_Employee emp ON emp.Employee_Id = tech.Employee_Id WHERE Service_Tech_Id = @service_tech_id
    SELECT @prevwage = Prevailing_Wage FROM OE_Job WHERE Job_Id = @job_id
    SET @rate_type = 'R'
    IF @prevwage > 0
    BEGIN
        SET @rate = @prevwage
        SET @rate_type = 'P'
    END
    SELECT @work_date = CAST(CONVERT ( char(10), @arrival_time, 102 ) as Datetime)
    SELECT @entered = GETDATE()
    IF @ts_from = 'A'
    BEGIN
        SET @units = CONVERT(money,DATEDIFF ( n , @arrival_time, @departure_time))/60
--        SELECT @units = DATEDIFF ( hour , @arrivaltime , @departure) + (((DATEDIFF ( n , @arrivaltime , @departure)) % 60
) * .01)
    END
    ELSE
    BEGIN
        SET @units = CONVERT(money,DATEDIFF ( n , @dispatch_time, @departure_time))/60
--        SELECT @units = DATEDIFF ( hour , @dispatchtime , @departure) + (((DATEDIFF ( n , @dispatchtime , @departure))
% 60) * .01)
    END
    SELECT @amount = @rate * @units

    IF @timesheet_id > 1
    BEGIN
        EXEC Job_Timesheet_UPD @timesheet_id, @installer_code, @work_date, @labortaskcode, @units, @rate,
        @amount, 'Dispatch Process Auto Created Based On Install Company', @posted_to_gl, @rate_type, @usercode,
        @entered
    END
    ELSE
    BEGIN
        EXEC Job_Timesheet_ADD @job_code, @job_task_id, @installer_code, @work_date, @labortaskcode, @units, @rate,
        @amount, 'Dispatch Process Auto Created Based On Install Company', @posted_to_gl, @usercode,
        @entered, @rate_type, 0, @timesheet_id OUTPUT
        UPDATE OE_Job_Dispatch SET Timesheet_ID = @timesheet_id WHERE Dispatch_Id = @dispatch_id
    END
END

    IF @service_tech_id > 1 AND @dispatch_time > 0
    BEGIN
        UPDATE OE_Job_Task
        SET Last_Service_Tech_Id = @service_tech_id, Last_Dispatch_Date = @dispatch_time
        WHERE Job_Task_Id = @job_task_id
    END
END
RETURN

```

