MsSQL - Restore Agent Jobs from Another MSDB database

This is to copy by specific Job

```
--- ErangaMSDB is the copy of the old msdb ---
DECLARE @JobID UNIQUEIDENTIFIER
DECLARE @SchedulerID Int
SELECT @JobID = job_id FROM ErangaMSDB.dbo.sysjobs WHERE NAME='Restore IPay'

--- Insert the jobs
INSERT msdb.dbo.sysjobs
SELECT * FROM ErangaMSDB.dbo.sysjobs
WHERE job_id=@JobID

--Insert the steps
INSERT msdb.dbo.sysjobsteps
SELECT * FROM ErangaMSDB.dbo.sysjobsteps
WHERE job_id=@JobID

--Insert the job history
SET IDENTITY_INSERT msdb.dbo.sysjobhistory ON
INSERT msdb.dbo.sysjobhistory
(instance_id,job_id,step_id,step_name,sql_message_id,sql_severity,
[message],run_status,run_date,run_time,run_duration,operator_id_emailed,
operator_id_netsent,operator_id_paged,retries_attempted,[server])
SELECT
instance_id,job_id,step_id,step_name,sql_message_id,sql_severity,
[message],run_status,run_date,run_time,run_duration,operator_id_emailed,
operator_id_netsent,operator_id_paged,retries_attempted,[server]
FROM ErangaMSDB.dbo.sysjobhistory
WHERE job_id=@JobID
```

```
SET IDENTITY_INSERT msdb.dbo.sysjobhistory OFF

--Insert the schedules
-- For the schedule - If more than 1 schedule, should add : WHERE schedule_id in (number1,number2)
SET IDENTITY_INSERT msdb.dbo.sysschedules ON

INSERT INTO msdb.dbo.sysschedules ( [schedule_id],[schedule_uid],[originating_server_id],[name],[owner_sid]
,[enabled],[freq_type],[freq_interval],[freq_subday_type],[freq_subday_interval]
,[freq_relative_interval],[freq_recurrence_factor],[active_start_date],[active_end_date]
,[active_start_time],[active_end_time],[date_created],[date_modified],[version_number]
)
SELECT [schedule_id],[schedule_uid],[originating_server_id],[name],[owner_sid]
,[enabled],[freq_type],[freq_interval],[freq_subday_type],[freq_subday_interval]
,[freq_relative_interval],[freq_recurrence_factor],[active_start_date],[active_end_date]
,[active_start_time],[active_end_time],[date_created],[date_modified],[version_number]
FROM ErangaMSDB.dbo.sysschedules
WHERE schedule_id =@SchedulerID

INSERT msdb.dbo.sysjobschedules
SELECT * FROM ErangaMSDB.dbo.sysjobschedules
WHERE job_id=@JobID
```

Came from Here

This is to copy all jobs

```
/**************************************Script to generate all the Jobs on a server
**************************************/--CreatedBy - Amit Mathur (v-amat)
--CreatedDate - 08/26/2009
/**************************************Script to generate all the Jobs on a server
**************************************/SET NOCOUNT ON
BEGIN TRY

PRINT 'USE [msdb]'
PRINT 'GO'
PRINT ''

DECLARE @JobID nvarchar(100),
        @JobName varchar (128),
        @JobCategory varchar (128),
```

```
        @JobCategoryClass varchar(128),
        @Now datetime,
        @Nowtext varchar(30)

    SELECT @Now = GETDATE()

    SELECT @Nowtext = CAST(@Now as varchar(30))



CREATE TABLE #Jobs (id int identity (1,1), jobid varchar(50))

INSERT INTO #Jobs (jobid) SELECT jobid = convert(varchar(50),job_id) FROM msdb.dbo.SysJobs WITH (NOLOCK)

DECLARE @MaxJobs int,
        @JobControl int
SELECT @JobControl = 1
SELECT @MaxJobs = MAX(id) FROM #jobs

--Create Jobs by looping through all the existing jobs on the server
WHILE (@JobControl <= @MaxJobs)
BEGIN --BEGIN Jobs

    SELECT @JobID = JobID FROM #jobs WHERE id = @JobControl

    SELECT @JobName = name FROM msdb.dbo.sysjobs_view WHERE Job_ID = @JobID
    SELECT @JobCategory =     sc.name, @JobCategoryClass = category_class FROM msdb.dbo.sysjobs sj
                        INNER JOIN msdb.dbo.syscategories sc
                        ON sc.category_id = sj.category_id
                        WHERE Job_ID = @JobID



    PRINT '/****** Object: Job ' + @JobName + ' Script Date:' + @Nowtext + ' ******/'
    PRINT 'IF EXISTS (SELECT job_id FROM msdb.dbo.sysjobs_view WHERE name = N''' + @JobName + ''')'
    PRINT 'EXEC msdb.dbo.sp_delete_job @job_name= N''' + @JobName + '''+ ', @delete_unused_schedule=1'
    PRINT 'GO'
    PRINT ''
    PRINT '/****** Object: Job ' + @JobName + ' Script Date:' + @Nowtext + ' ******/'
    PRINT 'BEGIN TRANSACTION'
    PRINT 'DECLARE @ReturnCode INT'
    PRINT 'SELECT @ReturnCode = 0'
```

```
    PRINT '/****** Object: JobCategory ' + QUOTENAME(@JobCategory) + ' Script Date:' + @Nowtext + ' ******/'
    PRINT 'IF NOT EXISTS (SELECT name FROM msdb.dbo.syscategories WHERE name = N''' + @JobCategory + '''
AND category_class = ' + @JobCategoryClass+ ')'
    PRINT 'BEGIN'
    PRINT 'EXEC @ReturnCode = msdb.dbo.sp_add_category @class=N''JOB'', @type=N''LOCAL'', @name = N''' +
@JobCategory + ''''
    PRINT 'IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback'
    PRINT ''
    PRINT 'END'
    PRINT ''
    PRINT 'DECLARE @jobId BINARY(16)'
    PRINT ''
    DECLARE @enabled int,
            @notify_level_eventlog int,
            @notify_level_email int,
            @notify_level_netsend int,
            @notify_level_page int,
            @delete_level int,
            @description nvarchar(128),
            @category_name nvarchar(128),
            @owner_login_name nvarchar(128),
            @notify_email_operator_name nvarchar(128)

    SELECT     @enabled = sj.enabled,
            @notify_level_eventlog = sj.notify_level_eventlog,
            @notify_level_email = sj.notify_level_email,
            @notify_level_netsend = sj.notify_level_netsend,
            @notify_level_page = sj.notify_level_page,
            @delete_level = sj.delete_level,
            @description = sj.[description],
            @category_name = sc.name,
            @owner_login_name = SUSER_NAME(sj.owner_sid),
            @notify_email_operator_name = so.name
    FROM msdb.dbo.sysjobs sj
    INNER JOIN msdb.dbo.syscategories sc
       ON sc.category_id = sj.category_id
    LEFT OUTER JOIN msdb.dbo.sysoperators so
       ON sj.notify_email_operator_id = so.id
    WHERE Job_ID = @JobID
```

```
    PRINT 'EXEC @ReturnCode = msdb.dbo.sp_add_job @job_name = N''' + @JobName + ''','
    PRINT '        @enabled=' + CAST(@enabled as varchar(30))+ ','
    PRINT '        @notify_level_eventlog=' + CAST(@notify_level_eventlog as varchar(30))+ ','
    PRINT '        @notify_level_email=' + CAST(@notify_level_email as varchar(30))+ ','
    PRINT '        @notify_level_netsend=' + CAST(@notify_level_netsend as varchar(30))+ ','
    PRINT '        @notify_level_page=' + CAST(@notify_level_page as varchar(30))+ ','
    PRINT '        @delete_level=' + CAST(@delete_level as varchar(30))+ ','
    PRINT '        @description=N''' + REPLACE(@description, '''','''''') + ''','
    PRINT '        @category_name=N''' + @category_name + ''','
    PRINT '        @owner_login_name=N''' + ISNULL(@owner_login_name,'sa') + ''','
IF @notify_email_operator_name IS NOT NULL
BEGIN
    PRINT '        @notify_email_operator_name=N''' + @notify_email_operator_name + ''', @job_id = @JobID
OUTPUT'
END
ELSE
    BEGIN
    PRINT '        @job_id = @JobID OUTPUT'
    END
    PRINT 'IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback'
    PRINT ''
    --CREATE STEPS
    DECLARE @MaxSteps int,
            @LoopControl int
    SELECT @LoopControl = 1
    SELECT @MaxSteps = MAX(step_id) FROM msdb.dbo.sysjobsteps WHERE Job_ID = @JobID

    WHILE (@LoopControl <= @MaxSteps)
        BEGIN
            DECLARE @step_name nvarchar (128),
                    @step_id int,
                    @cmdexec_success_code int,
                    @on_success_action int,
                    @on_success_step_id int,
                    @on_fail_action int,
                    @on_fail_step_id int,
                    @retry_attempts int,
                    @retry_interval int,
                    @os_run_priority int,
                    @subsystem nvarchar (128),
```

```
        @command nvarchar (max),
        @database_name nvarchar(128),
        @flags int

    SELECT     @step_name = step_name,
        @step_id = step_id,
        @cmdexec_success_code = cmdexec_success_code,
        @on_success_action = on_success_action,
        @on_success_step_id = on_success_step_id,
        @on_fail_action = on_fail_action,
        @on_fail_step_id = on_fail_step_id,
        @retry_attempts = retry_attempts,
        @retry_interval = retry_interval,
        @os_run_priority = os_run_priority,
        @subsystem = subsystem,
        @command = command,
        @database_name = database_name,
        @flags = flags
    FROM msdb.dbo.sysjobsteps WHERE Job_ID = @JobID
    AND step_id = @LoopControl
    PRINT ''
    PRINT '/****** Object: Step ' + @step_name + ' Script Date: ' + @Nowtext + '******/'

    PRINT 'EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId, @step_name=N''' +
@step_name + ''','
    PRINT '       @step_id=' + CAST(@step_id as varchar(30))+ ','
    PRINT '       @cmdexec_success_code=' + CAST(@cmdexec_success_code as varchar(30))+ ','
    PRINT '       @on_success_action=' + CAST(@on_success_action as varchar(30))+ ','
    PRINT '       @on_success_step_id=' + CAST(@on_success_step_id as varchar(30))+ ','
    PRINT '       @on_fail_action=' + CAST(@on_fail_action as varchar(30))+ ','
    PRINT '       @on_fail_step_id=' + CAST(@on_fail_step_id as varchar(30))+ ','
    PRINT '       @retry_attempts=' + CAST(@retry_attempts as varchar(30))+ ','
    PRINT '       @retry_interval=' + CAST(@retry_interval as varchar(30))+ ','
    PRINT '       @os_run_priority=' + CAST(@os_run_priority as varchar(30))+ ', @subsystem=N''' +
@subsystem + ''','
    PRINT '       @command=N''' + REPLACE(@command, '''','''''') + ''','
    PRINT '       @database_name=N''' + @database_name + ''','
    PRINT '       @flags=' + CAST(@flags as varchar(30))
    PRINT ''
    SELECT @LoopControl = @LoopControl + 1
```

```
    END -- End Steps While
PRINT ''
PRINT 'IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback'
PRINT 'EXEC @ReturnCode = msdb.dbo.sp_update_job @job_id = @jobId, @start_step_id = 1'
PRINT 'IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback'

PRINT ''

--CREATE SCHEDULES

DECLARE @MaxSchedules int,
        @SchedulesLoopControl int
SELECT @SchedulesLoopControl = 1

CREATE TABLE #Schedules (id int identity (1,1), schedule_id int)

INSERT INTO #Schedules (schedule_id) SELECT schedule_id = sjs.schedule_id
FROM msdb.dbo.sysjobschedules sjs WITH (NOLOCK)
--INNER JOIN msdb.dbo.sysschedules ss WITH (NOLOCK) ON sjs.schedule_id = ss.schedule_id
WHERE sjs.Job_ID = @JobID

SELECT @MaxSchedules = MAX(id) FROM #Schedules

IF EXISTS (SELECT COUNT(*) FROM #Schedules)
BEGIN
    WHILE (@SchedulesLoopControl <= @MaxSchedules)
        BEGIN
            DECLARE @name nvarchar(2000),
                    @sch_enabled int,
                    @freq_type int,
                    @freq_interval int,
                    @freq_subday_type int,
                    @freq_subday_interval int,
                    @freq_relative_interval int,
                    @freq_recurrence_factor int,
                    @active_start_date int,
                    @active_end_date int,
                    @active_start_time int,
                    @active_end_time int,
                    @schedule_uid nvarchar (50)
```

```
SELECT     @name = name,
        @sch_enabled = enabled,
        @freq_type = freq_type,
        @freq_interval = freq_interval,
        @freq_subday_type = freq_subday_type,
        @freq_subday_interval = freq_subday_interval,
        @freq_relative_interval = freq_relative_interval,
        @freq_recurrence_factor = freq_recurrence_factor,
        @active_start_date = active_start_date,
        @active_end_date = active_end_date,
        @active_start_time = active_start_time,
        @active_end_time = active_end_time,
        @schedule_uid = schedule_uid
        FROM msdb.dbo.sysjobschedules sjs WITH (NOLOCK)
        INNER JOIN msdb.dbo.sysschedules ss WITH (NOLOCK) ON sjs.schedule_id = ss.schedule_id
        INNER JOIN #Schedules s ON ss.schedule_id = s.schedule_id
        WHERE sjs.Job_ID = @JobID
        AND s.id = @SchedulesLoopControl

    PRINT 'EXEC @ReturnCode = msdb.dbo.sp_add_jobschedule @job_id=@jobId, @name=N''' +
REPLACE(@name, '''','''''') + ''','
    PRINT '       @enabled=' + CAST(@sch_enabled as varchar(30))+ ','
    PRINT '       @freq_type=' + CAST(@freq_type as varchar(30))+ ','
    PRINT '       @freq_interval=' + CAST(@freq_interval as varchar(30))+ ','
    PRINT '       @freq_subday_type=' + CAST(@freq_subday_type as varchar(30))+ ','
    PRINT '       @freq_subday_interval=' + CAST(@freq_subday_interval as varchar(30))+ ','
    PRINT '       @freq_relative_interval=' + CAST(@freq_relative_interval as varchar(30))+ ','
    PRINT '       @freq_recurrence_factor=' + CAST(@freq_recurrence_factor as varchar(30))+ ','
    PRINT '       @active_start_date=' + CAST(@active_start_date as varchar(30))+ ','
    PRINT '       @active_end_date=' + CAST(@active_end_date as varchar(30))+ ','
    PRINT '       @active_start_time=' + CAST(@active_start_time as varchar (30)) + ','
    PRINT '       @active_end_time=' + CAST(@active_end_time as varchar (30)) + ','
    PRINT '       @schedule_uid=N''' + @schedule_uid + ''''
    PRINT ''

    PRINT ''
    PRINT 'IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback'
    PRINT ''
```

```
            SELECT @SchedulesLoopControl = @SchedulesLoopControl + 1
        END -- End Schedules While loop
    END -- END IF (SELECT COUNT(*) FROM #Schedules) > 0

    DECLARE @server_name varchar(30)
    SELECT @server_name = CASE server_id WHEN 0 THEN 'local' ELSE 'Multi-Server' END
    FROM msdb.dbo.sysjobservers WHERE Job_ID = @JobID
    PRINT 'EXEC @ReturnCode = msdb.dbo.sp_add_jobserver @job_id = @jobId, @server_name =N''(' +
@server_name + ')'''
    PRINT 'IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback'
    PRINT 'COMMIT TRANSACTION'
    PRINT 'GOTO EndSave'
    PRINT 'QuitWithRollback:'
    PRINT ' IF (@@TRANCOUNT > 0) ROLLBACK TRANSACTION'
    PRINT 'EndSave:'
    PRINT ''
    PRINT 'GO'
    PRINT ''
    PRINT ''

    SELECT @JobControl = @JobControl + 1

    DROP TABLE #Schedules

END --End Jobs

DROP TABLE #Jobs

END TRY

BEGIN CATCH
    DROP TABLE #Jobs
    DROP TABLE #Schedules
END CATCH;
```

Came from here

Revision #2
Created 11 September 2025 16:52:18 by Steve Ling
Updated 12 September 2025 12:45:38 by Steve Ling