Create a user for AlwaysOn Servers

* On the primary server
  + Create the user.
  + Do not assign any DB rights yet
  + Open a Query window on the Master database
  + Create Stored procedure **sp\_help\_revlogin**
    - <https://support.microsoft.com/en-za/help/918992/how-to-transfer-logins-and-passwords-between-instances-of-sql-server>
  + EXEC sp\_help\_revlogin
  + Copy the “create user” script form the output
* On the secondary server
  + Open a Query window on the Master database
  + Run the copied “create user” script
* On the primary server
  + Grant rights to your user

USE master

GO

IF OBJECT\_ID ('sp\_hexadecimal') IS NOT NULL

DROP PROCEDURE sp\_hexadecimal

GO

CREATE PROCEDURE sp\_hexadecimal

@binvalue varbinary(256),

@hexvalue varchar (514) OUTPUT

AS

DECLARE @charvalue varchar (514)

DECLARE @i int

DECLARE @length int

DECLARE @hexstring char(16)

SELECT @charvalue = '0x'

SELECT @i = 1

SELECT @length = DATALENGTH (@binvalue)

SELECT @hexstring = '0123456789ABCDEF'

WHILE (@i <= @length)

BEGIN

DECLARE @tempint int

DECLARE @firstint int

DECLARE @secondint int

SELECT @tempint = CONVERT(int, SUBSTRING(@binvalue,@i,1))

SELECT @firstint = FLOOR(@tempint/16)

SELECT @secondint = @tempint - (@firstint\*16)

SELECT @charvalue = @charvalue +

SUBSTRING(@hexstring, @firstint+1, 1) +

SUBSTRING(@hexstring, @secondint+1, 1)

SELECT @i = @i + 1

END

SELECT @hexvalue = @charvalue

GO

IF OBJECT\_ID ('sp\_help\_revlogin') IS NOT NULL

DROP PROCEDURE sp\_help\_revlogin

GO

CREATE PROCEDURE sp\_help\_revlogin @login\_name sysname = NULL AS

DECLARE @name sysname

DECLARE @type varchar (1)

DECLARE @hasaccess int

DECLARE @denylogin int

DECLARE @is\_disabled int

DECLARE @PWD\_varbinary varbinary (256)

DECLARE @PWD\_string varchar (514)

DECLARE @SID\_varbinary varbinary (85)

DECLARE @SID\_string varchar (514)

DECLARE @tmpstr varchar (1024)

DECLARE @is\_policy\_checked varchar (3)

DECLARE @is\_expiration\_checked varchar (3)

DECLARE @defaultdb sysname

IF (@login\_name IS NULL)

DECLARE login\_curs CURSOR FOR

SELECT p.sid, p.name, p.type, p.is\_disabled, p.default\_database\_name, l.hasaccess, l.denylogin FROM

sys.server\_principals p LEFT JOIN sys.syslogins l

ON ( l.name = p.name ) WHERE p.type IN ( 'S', 'G', 'U' ) AND p.name <> 'sa'

ELSE

DECLARE login\_curs CURSOR FOR

SELECT p.sid, p.name, p.type, p.is\_disabled, p.default\_database\_name, l.hasaccess, l.denylogin FROM

sys.server\_principals p LEFT JOIN sys.syslogins l

ON ( l.name = p.name ) WHERE p.type IN ( 'S', 'G', 'U' ) AND p.name = @login\_name

OPEN login\_curs

FETCH NEXT FROM login\_curs INTO @SID\_varbinary, @name, @type, @is\_disabled, @defaultdb, @hasaccess, @denylogin

IF (@@fetch\_status = -1)

BEGIN

PRINT 'No login(s) found.'

CLOSE login\_curs

DEALLOCATE login\_curs

RETURN -1

END

SET @tmpstr = '/\* sp\_help\_revlogin script '

PRINT @tmpstr

SET @tmpstr = '\*\* Generated ' + CONVERT (varchar, GETDATE()) + ' on ' + @@SERVERNAME + ' \*/'

PRINT @tmpstr

PRINT ''

WHILE (@@fetch\_status <> -1)

BEGIN

IF (@@fetch\_status <> -2)

BEGIN

PRINT ''

SET @tmpstr = '-- Login: ' + @name

PRINT @tmpstr

IF (@type IN ( 'G', 'U'))

BEGIN -- NT authenticated account/group

SET @tmpstr = 'CREATE LOGIN ' + QUOTENAME( @name ) + ' FROM WINDOWS WITH DEFAULT\_DATABASE = [' + @defaultdb + ']'

END

ELSE BEGIN -- SQL Server authentication

-- obtain password and sid

SET @PWD\_varbinary = CAST( LOGINPROPERTY( @name, 'PasswordHash' ) AS varbinary (256) )

EXEC sp\_hexadecimal @PWD\_varbinary, @PWD\_string OUT

EXEC sp\_hexadecimal @SID\_varbinary,@SID\_string OUT

-- obtain password policy state

SELECT @is\_policy\_checked = CASE is\_policy\_checked WHEN 1 THEN 'ON' WHEN 0 THEN 'OFF' ELSE NULL END FROM sys.sql\_logins WHERE name = @name

SELECT @is\_expiration\_checked = CASE is\_expiration\_checked WHEN 1 THEN 'ON' WHEN 0 THEN 'OFF' ELSE NULL END FROM sys.sql\_logins WHERE name = @name

SET @tmpstr = 'CREATE LOGIN ' + QUOTENAME( @name ) + ' WITH PASSWORD = ' + @PWD\_string + ' HASHED, SID = ' + @SID\_string + ', DEFAULT\_DATABASE = [' + @defaultdb + ']'

IF ( @is\_policy\_checked IS NOT NULL )

BEGIN

SET @tmpstr = @tmpstr + ', CHECK\_POLICY = ' + @is\_policy\_checked

END

IF ( @is\_expiration\_checked IS NOT NULL )

BEGIN

SET @tmpstr = @tmpstr + ', CHECK\_EXPIRATION = ' + @is\_expiration\_checked

END

END

IF (@denylogin = 1)

BEGIN -- login is denied access

SET @tmpstr = @tmpstr + '; DENY CONNECT SQL TO ' + QUOTENAME( @name )

END

ELSE IF (@hasaccess = 0)

BEGIN -- login exists but does not have access

SET @tmpstr = @tmpstr + '; REVOKE CONNECT SQL TO ' + QUOTENAME( @name )

END

IF (@is\_disabled = 1)

BEGIN -- login is disabled

SET @tmpstr = @tmpstr + '; ALTER LOGIN ' + QUOTENAME( @name ) + ' DISABLE'

END

PRINT @tmpstr

END

FETCH NEXT FROM login\_curs INTO @SID\_varbinary, @name, @type, @is\_disabled, @defaultdb, @hasaccess, @denylogin

END

CLOSE login\_curs

DEALLOCATE login\_curs

RETURN 0

GO