

Changing the Service User Account

Last Modified on 03/19/2020 11:28 am EDT

V9.0

Description

Use this script to change the identity that runs a Cora SeQuence service application.

Prerequisite

Make sure that the identity that you set for BRS or JES is a valid user in Cora SeQuence and has the required permissions.

Procedure

1. In the Get-CoraSeQuenceService, replace the following parameters:
 - a. Service: enter the service type for which you need to set the identity: ADSS, BRS, or JES.
 - b. ServiceID: enter the ID of the service.

NOTE: ServiceID is required only if there is more than one instance of the service.

2. Run the script
3. When prompted, enter the relevant user credentials.

NOTE

After you change a JES account, you need to register JES again.
For details, see [this article](#).

Sample

```

#Get the service
$service = Get-CoraSeQuenceService -Service BRS
if (-not ($service))
{
    Write-Error -Message "Service not found!"
}
else
{
    #Get the credentials to use for this service account
    $serviceAccount = Get-Credential
    #Stop the service if it is running
    $state = $service.State
    if ($state -eq "Running")
    {
        $result = $service.StopService()
        if ($result.ReturnValue -ne 0)
        {
            Write-Error -Message "Stop service failed with return value $($result.ReturnValue). For more information see https://docs.microsoft.com/en-us/windows/desktop/cimwin32prov/stopservice-method-in-class-win32-service#return-value"
            exit
        }
    }
    #Set the service account identity
    $result = $service.Change($null,$null,$null,$null,$null,$null,$($serviceAccount.UserName),$($serviceAccount.GetNetworkCredential().Password),$null,$null,$null)
    if ($result.ReturnValue -ne 0)
    {
        Write-Error -Message "Change of service account identity failed with return value $($changeResult.ReturnValue). For more information see https://docs.microsoft.com/en-us/windows/desktop/cimwin32prov/change-method-in-class-win32-service#return-value"
    }
    #Start the service if it was running before
    if ($state -eq "Running")
    {
        $result = $service.StartService()
        if ($result.ReturnValue -ne 0)
        {
            Write-Error -Message "Start service failed with return value $($result.ReturnValue). For more information see https://docs.microsoft.com/en-us/windows/desktop/cimwin32prov/startservice-method-in-class-win32-service#return-value"
            exit
        }
    }
}
}

```

NOTE

This sample script is suitable for a single instance of a service only.