SAP Netweaver Applications

Silent Install Process

# Purpose of the SAP Silent Installation Process

The SAP Business Suites running on Netweaver include applications of many types. E.g., Enterprise Core Components (ECC), Supply Chain Management (SCM), Customer Relationship Management (CRM) to name a few. Also adding on the variety of options on the intended infrastructure architecture, e.g., Central build (aka 2-tier) and Distributed build (3-tier) result in many installation variations. The customary installation process has been using the SAP installer (SAPInst.exe) or the newer SAP Software Provisioning Manager (SWPM) which is based on SAPInst. The silent install process hasn’t been widely adopted. Partners and customers have been accustomed to spending many hours or days installing an SAP system from the ground up interactively.

This document describes the SAP silent installation process with the intention for it to be invoked by a PowerShell script as part of a post infrastructure automation process which provisions the Azure network, compute, and storage. The PowerShell invocation of the SAP silent install process makes possible the end-to-end build automation of an SAP Business Suites on Netweaver application on Microsoft Azure. This end-to-end SAP system installation process or ‘template’ can be recreated for many SAP Business Suites applications for repeated installations thereby greatly reducing the time required to provision an SAP system.

The process described here leverages the SAP Software Provisioning Manager (SWPM). Much like the interactive installation, every server type (central build or distributed build for an application server, or database server) requires a separate installation pass to generate an installation template for one SAP system.

# Setting up a Good Base for the Silent Install

The author’s experience in setting up a good base for an SAP silent install is a successful interactive installation. A successful interactive install validates that the SAP installation media has been correctly downloaded and placed at a location reachable by SAPinst. It also is the time when the SAP engineer resolves installation issues. Also, the interactive installation provides the opportunity to monitor the resource consumption (e.g. CPU) for a given Azure VM configuration during database load so one can adjust the number of load processes to optimize the silent installation run time. Time saved here is multiplied by the number of times the same template is used for subsequent installations.

Other important prerequisites of a successful interactive installation include but are not limited to the following:

* All necessary OS or DB patches/hot fixes
* Set page-file to required size

# Key elements of an SAP silent install

There are a few required items to setup an SAP silent install:

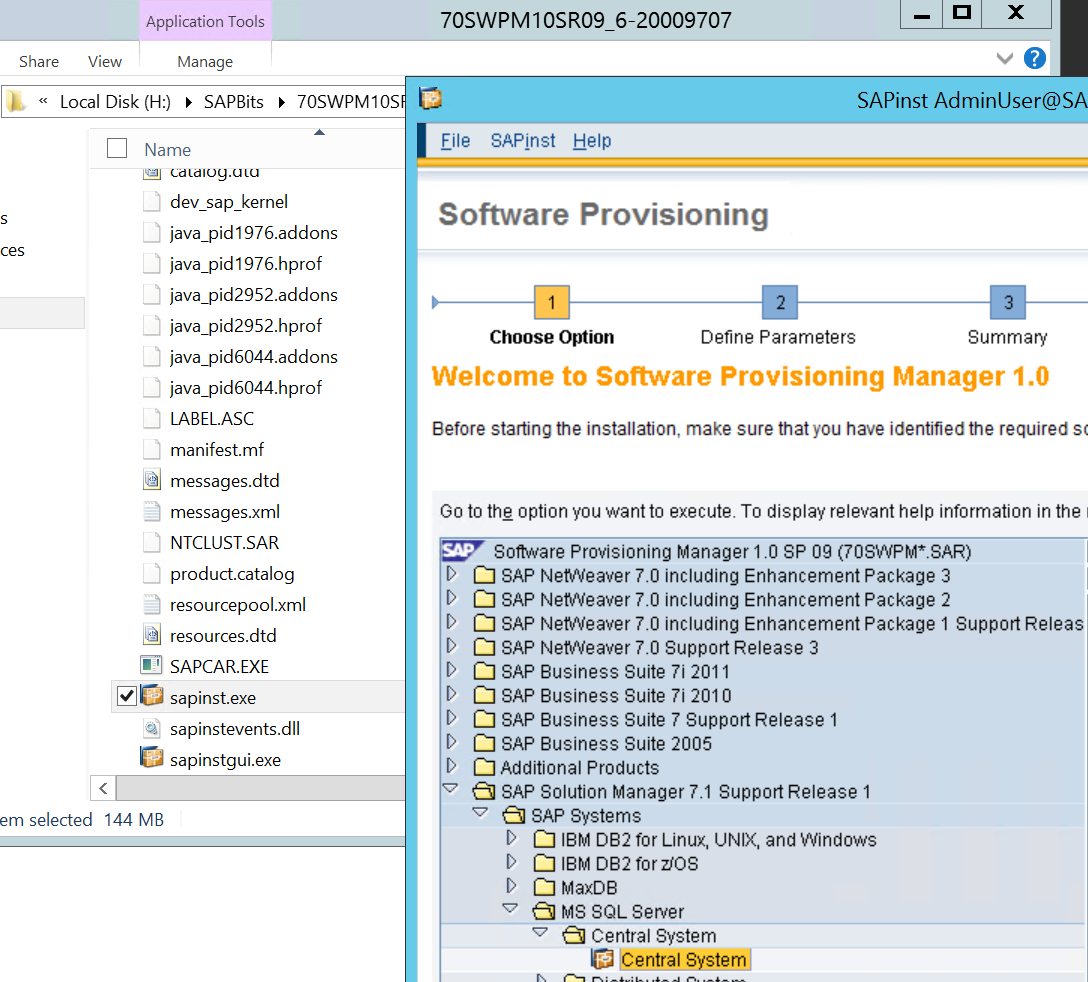
1. The inifile.xml (ie: ‘installation template’), is the file generated by the SAPinst interactive installation process. This file captures the SAP Basis engineer’s entries to define the configuration for the SAP installation, such as what the number of database load processes should be used, user’s passwords, etc. An important piece of information for the silent install is the PRODUCT\_ID of the application being installed (ECC for example) which is captured here. This file is specific to the target server configuration and the SAP application being installed. This installation template can only be used on servers with identical configuration (e.g., disks and CPU)
2. The Start\_dir.cd file is a text file listing all the paths to the installation media. A key point about these paths is they need to lead the SAPinst tool to sub-folders containing the media’s CD labels. The interactive installation mentioned earlier helps validate these paths. This file is created manually by the SAP Basis engineer
3. Keydb.dtd – Doc type definition for keybd.xml file, tables definition and content needed during the service execution. This file came from the extraction of the SAP installation media
4. Doc.dtd – Doc type definition used by SAPinst documentation. This file came from the extraction of the SAP installation media

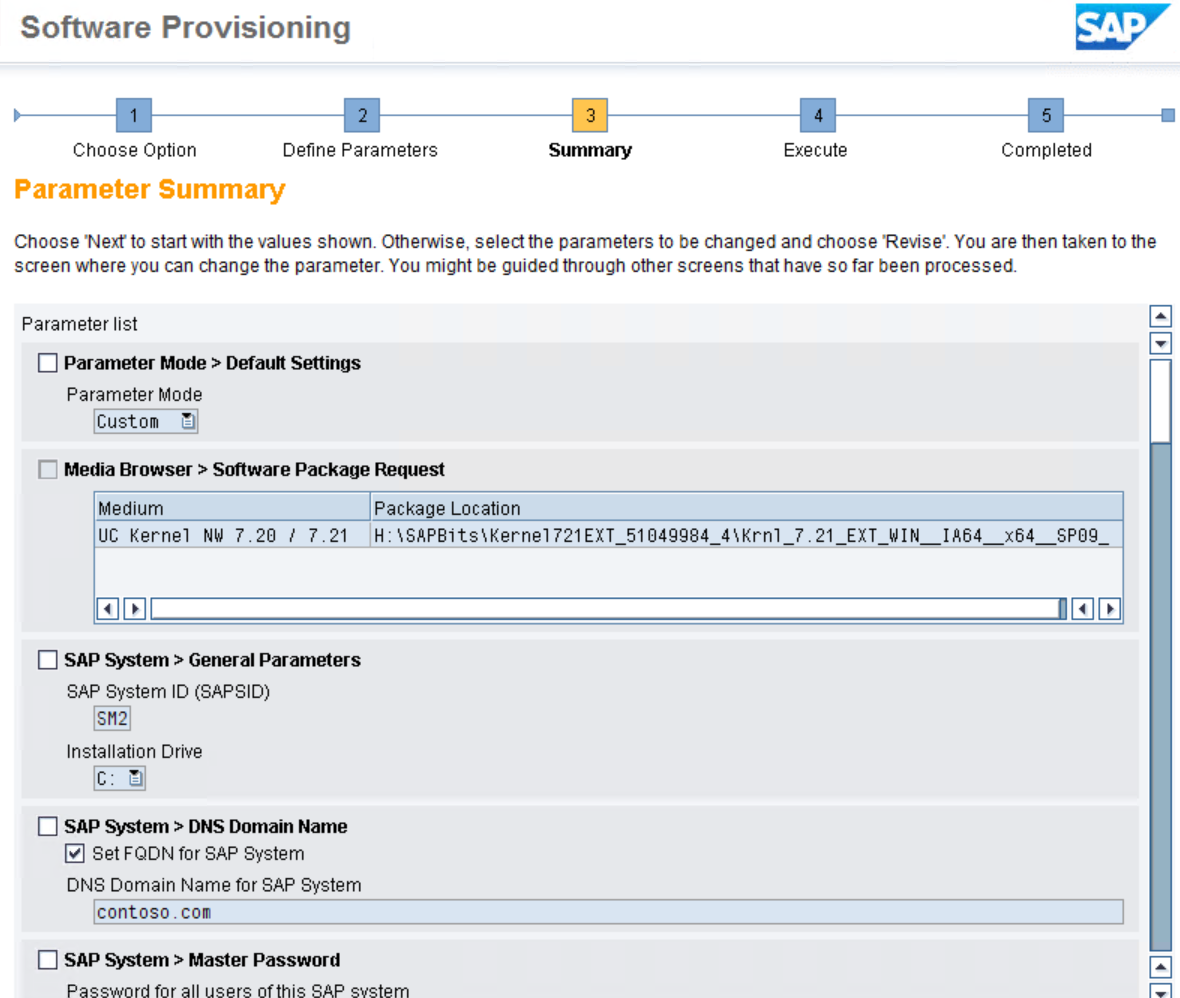
These four files can be placed in any directory intended to be used as the working directory of the installation. The recommendation is to create a new folder for the silent install.

# The silent install setup process

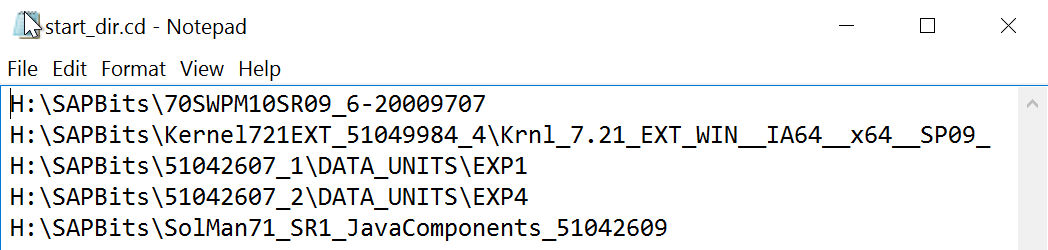
The SAP silent install process is simple.

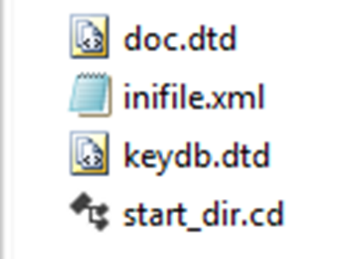
1. Step through SAPinst interactively to generate the inifile.xml. The below snapshot is the interactive install process for a Solution Manager Central System which will install all SAP components like the Central Services, Application Server, and the Database Server in one VM.





At the point of the SAPInst Summary screen, stop the interaction with SAPinst at the Summary screen and copy the inifile.xml file from your SAP installation directory by default located at C:\Program Files\sapinst\_instdir and drop it in your silent install directory. Then terminate SAPinst and ensure it completely exist by checking on Task Manager

1. Create the start\_dir.cd file by listing all paths to install media. This step is best done during the interaction with SAPinst while the SAP installation media paths are being supplied and copied onto the text file. Here’s an example of the file 
2. Create an install directory at any location with any name
3. Pick up the doc.dtd and the keydb.dtd files from your interactive install directory
4. Drop the inifile.xml, doc.dtd, keydb.dtd, start\_dir.cd files onto the install directory just created
5. Find the PRODUCT\_ID from the inifile.xml embedded in the installation service statement at the beginning of the file (tail- end of a long string)
6. At this point, If your silent install directory contain the four files (we refer to these as the ‘seed files’ like we show below, you have all the necessary components to start the SAP silent install.



To repeat the same installation on a VM with identical configuration,

# Running the SAP Silent install

1. Change directory to the install folder you created
2. Set PATH variable to include the Software Deployment Manager (SWPM) so the SAPinst command can be found or specify the explicit path to the SAPinst command
3. Execute SAPinst with variables:
   * SAPINST\_PARAMETER\_CONTAINER\_URL=inifile.xml
   * SAPINST\_EXECUTE\_PRODUCT\_ID (found in inifile.xml)
   * SAPINST\_SKIP\_DIALOGS=true
   * -nogui
   * -noguiserver

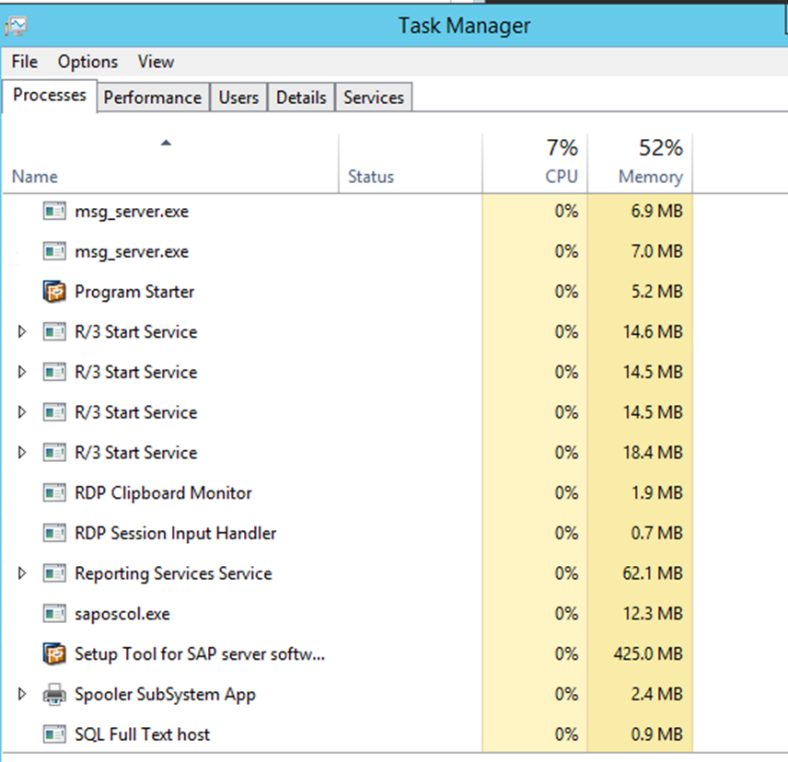
Example -> c:\Silent>sapinst SAPINST\_PARAMETER\_CONTAINER\_URL=inifile.xml SAPINST\_EXECUTE\_PRODUCT\_ID=NW\_Onehost:SOLMAN71.MSS.PD SAPINST\_SKIP\_DIALOGS=true -nogui -noguiserver

# Monitoring the SAP Silent Install

The monitoring of the SAP silent install is done with the same tools for an interactive install. Below are some common tools used for such an installation

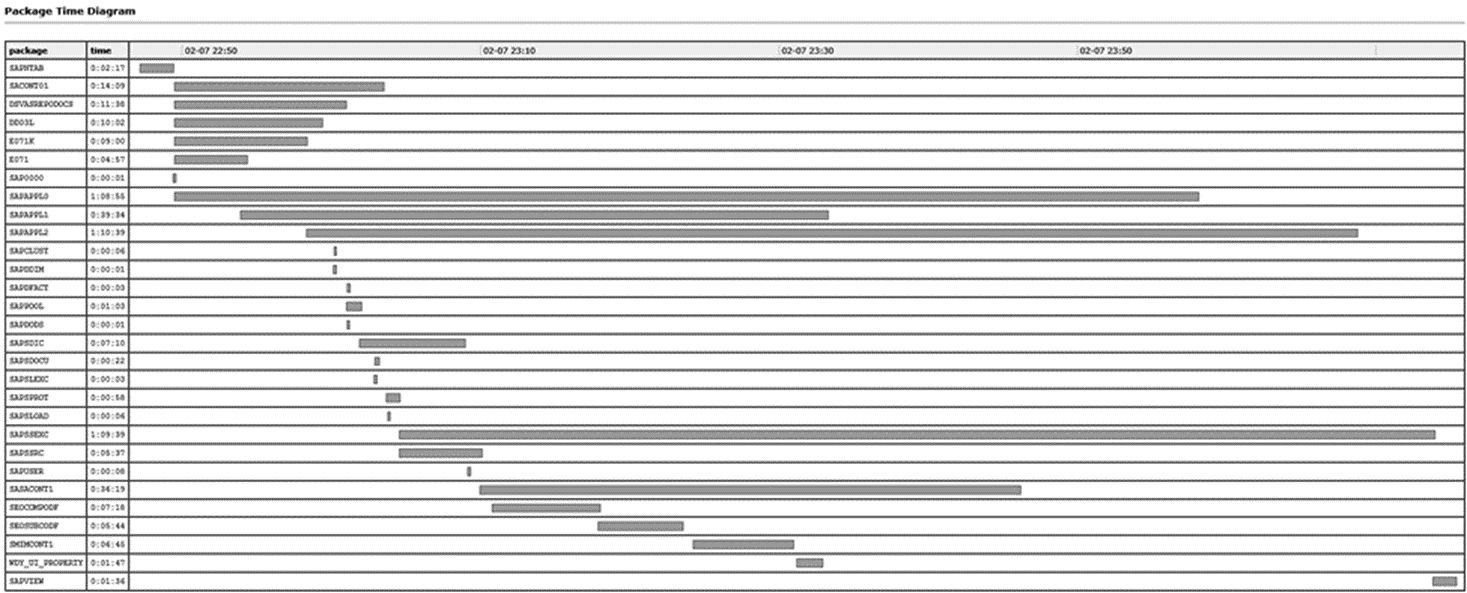
* + Logs are written in the install directory where you launch the sapinst command
  + SAPinst.log
  + SAPinst\_dev.log
  + Individual SAP content package imports logs
  + Task manager

At installation run time, the Program Starter and the Setup Tool for SAP Software should be present and active most of the time. Also, the presence of multiple R3load processes is a progress indication. Near the completion of the installation, there should be multiple disp+work processes for the ABAP stack, a jcontrol, and one or two jlaunch processes for the Java stack if the application you install includes the Java application server.



# Measuring Run Time

The SAP Migration Time Analyzer (MigTime) reads the install logs and render a graphical chart of the resultant load time for each of the SAP packages. Review the SAP Note 784118 to learn details about the tool. The snapshot below shows an example of a 2-tier Solution Manager system. The database load time alone take around 3 hours.



# When you need to clean up

Often time, things don’t work the way they are planned. When there is a need to clean up the VM, here are a few steps:

1. Log in as user <sid>adm
2. Use SAPinst on the SWMP media
3. Select <product>, software life-cycle options, Uninstall

Attempting to manually clean out the registry and removing software from the install folder generally don’t end well.

# Restarting silent install on the same VM after a failed attempt

Whether cleanup is required depends on the point where the prior installation failed. In other words, the system state at the time of installation failure.

1. Running with the same options W/O clean-up may lead to
   * unknown message ID (syslib.filesystem.creatingFile) with parameter(s): C:\Users\sapadmin\AppData\Local\Temp\2\sapinst\_exe.4344.1454978029\dev\_sap\_kernel\_test\_09\_Feb\_2016\_00\_34\_15. Aborted
   * Changed the SID and rerun resulted in the same error

In the situations described above, spending the time to run the SAP uninstall tool maybe the most prudent action.

# Restarting the silent install on the same VM after cleanup

After a failed attempt of SAP silent install and cleaning up the VM, repeating the install on the same install directory is fine. If starting out fresh is desired, one can do the following:

1. Remove or rename old install directory while ensuring the new install directory contains the four ‘seed files’
2. Change directory to the install directory
3. Restart SAPinst command with the same parameters as it was run the first time

# Repeating the installation on a different VM

This is the good scenario when all went well with the SAP silent install template. Now you have a need to repeat the same installation on possibly a few VMs.

1. Copy the install directory with only the 4 seed files
2. Ensure target host has identical disk structure and software repository structure
3. Edit the inifile.xml to replace the old <sid> with the new one throughout
4. Change fld name=“TransHost“ strval to your new hostname if the new host is your transport host. Else, keep or change TransHost value to the hostname of the transport host
5. Replace all hostname fields with new host (domain, httphost etc)
6. Run the SAPinst command with the parameters described in the ‘Running the SAP Silent install’ section