vmware

SOA Services - Monitoring

Technical Specifications Document

Table of Contents

1. Introduction – SOA Services Monitoring

Oracle SOA Suite is a comprehensive, hot-pluggable software suite that enables you to build, deploy, and manage integrations using service-oriented architecture (SOA). Oracle SOA Suite provides the following capabilities:

- Consistent tooling
- A single deployment and management model
- End-to-end security
- Unified metadata management

SOA uses an *Enterprise Service Bus* (ESB) to handle communication internally between the various services that make up an application, and externally with other applications and clients. The ESB plays a central role in the functioning of an SOA application. It enables diversity in the protocols that services use to communicate with each other. It acts as a central mediator between all services and aims to provide out-of-the-box support for various protocols and, in this way, quicken communication.

SOA 12c Strengths

- Improved Runtime Visibility
- End-to-end visibility of components
- Highly responsive Enterprise Manager
- Unified fault management experience
- New Error Hospital
- Developer Productivity
- Debugging and testing capabilities in Jdeveloper
- Faster Startup Time and Optimize Memory Usage
- New Adapters – SAP, SFDC, Eloqua
- Roadmap for Continuous Delivery

SOA monitoring revolves around the processing of messages between services. The ESB is the layer that transmits messages between services, so if a service is running slowly or has failed, the number of messages to that service pile up at the ESB. Therefore, it's important to track the number of messages in the queue to spot latency or failures in services. Additionally, for each service, it helps to track the messages processed per minute or five minutes and monitor any deviations from the average.

2. Monitoring

Oracle SOA Suite 12c administrator typically focuses on 3 areas while monitoring they are:

- Transactions
- Instance state and performance
- Infrastructure

2.1 Transaction Monitoring

The responsibilities of an administrator is to ensure that the infrastructure executes transactions reliably and efficiently by providing complete details on all aspects of the environment.

Transaction monitoring involves the following:

- Reviewing faulted instances to act (retry, replay, or ignore).
- Searching log files for additional log information on faulted instances.
- Searching through composite sensors if the end-user complains of a business transaction not going through (if composite sensors are implemented in the code).
- Enabling selective tracing, which allows you to change the trace level for a defined scope. Examples of scope are a logged-in user, deployed application, or BPEL composite.

When monitoring instances, the goal is:

- To identify transactions that are not completed successfully.
- To determine further action and ensure that the transactions do not experience poor performance.

When a message is received by the SOA Infrastructure, it may pass through multiple components within your infrastructure and may even traverse multiple external systems as well. For example, an order may be received by an OSB service which then passes it on to a BPEL process for further processing before finally placing it into a queue. Afterwards, it may be consumed by a third-party application that processes this order before sending it back to a Mediator service that routes it to the final order management application.

If one of the six steps in this integration fails, how can you identify the location of the message? It would also be important to know the duration of execution by the component to determine whether they are within the defined SLAs.

2.2. Instance State and Performance Monitoring

Monitoring instance state and performance involves the following:

- Reviewing the performance summary and request processing pages on the console to graphically display specific metrics on selected composites.
- Running SQL queries to retrieve summary and detailed performance information on composite instances.

2.3 Infrastructure Monitoring

Infrastructure monitoring involves the following:

- Reviewing filesystem log files for system and application errors. monitoring the Oracle WebLogic Server managed servers for the overall health.
- Monitoring the JVM for appropriate sizing and garbage collection frequency.
- Monitoring Java Message Service (JMS) destinations, such as queues and topics to ensure that messages are being processed.
- Monitoring data sources to pre-emptively identify any issues.
- Monitoring threads.
- Monitoring operating to system-level parameters.

3. Faulted Instances

One of the more common activities that an administrator performs is retrieving a list of faulty or rejected SOA composite instances and get the necessary information to troubleshoot them. In SOA Suite 12c, this is simplified.

Follow the procedure to access the new fault screen:

- Click on **SOA-infra** -> **Error Hospital tab**.
- Click on **Review** and **Recover** the faulted instances.
- **Search** the faults based upon instance start times and fault times.

Additionally, they can be grouped by Composite, Partition, and Fault Code to name a few.

This helps in getting a quick glance at issues that occur in the infrastructure and sometimes provides the ability to act on them.

Unfortunately, the fault shown on the console may not contain enough information to effectively handle the error and a review of the log files are necessary.

4. Searching Composite Sensors

Composite sensors are added to the SOA composite at design time by the developer. They are not specific to BPEL or Mediator but are instead captured at the composite level. They provide a method of implementing traceable fields on messages.

For example, composite sensors can barely be used to capture business indicators, such as a customer ID or an order number, and persist this data in the database, which then becomes searchable through the Fusion Middleware Control console.

If the composite is not designed to capture composite sensors, you will not be able to search on them.

To search for a composite sensor, perform the following steps:

- Click on the **Flow Instances** tab and confirm that the Flow Instance header exists.
- If the Flow Instance header exists do not then click on **Add/Remove Filters**.
- Check the **Flow Instance** box, and click on the **OK** tab.
- Click on the **Add Sensor Values** tab and choose the **Sensor Name field** and corresponding value.
- Select a new search field this allows composite sensor on the page.

Note: *Remember, composite sensors can technically capture any type of data defined by the developer at design time. When searching by a composite sensor, a list of instances are retrieved that contain that sensor value.*

5. Monitoring the Environments

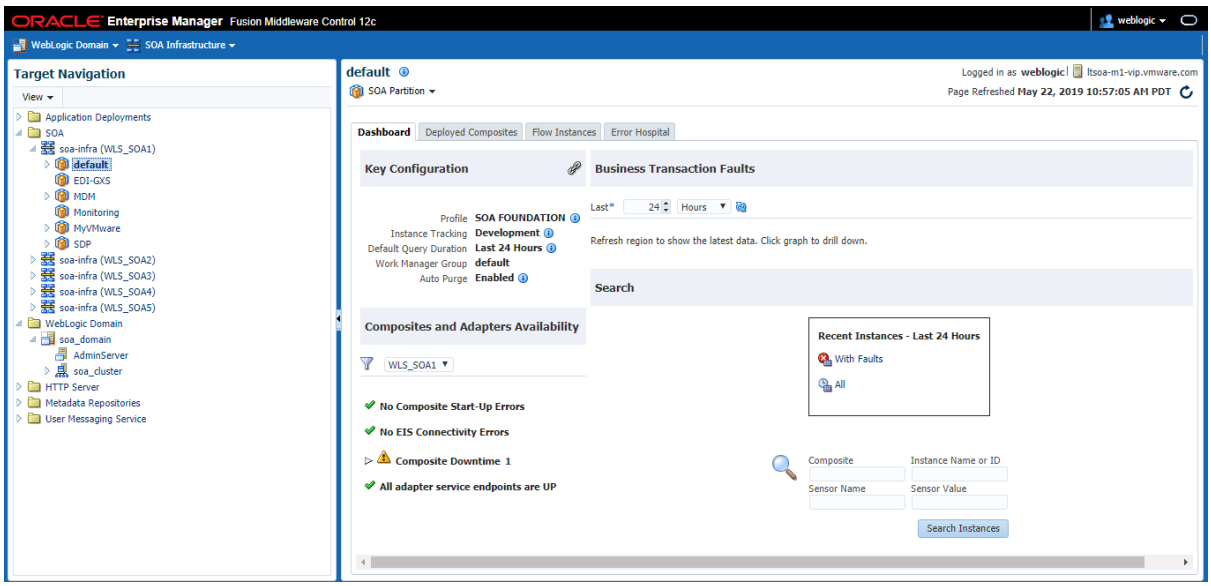
5.1 Monitoring Overall Portion of SOA

Monitor the overall status of your environment from the Dashboard pages of the SOA Partition or an individual partition.

Click to Home Dashboard.

1. Expand SOA.
2. Navigate to soa-infra (WLS_SOA1).
3. Select default partition.

The SOA Partition Dashboard page displays the following Details.

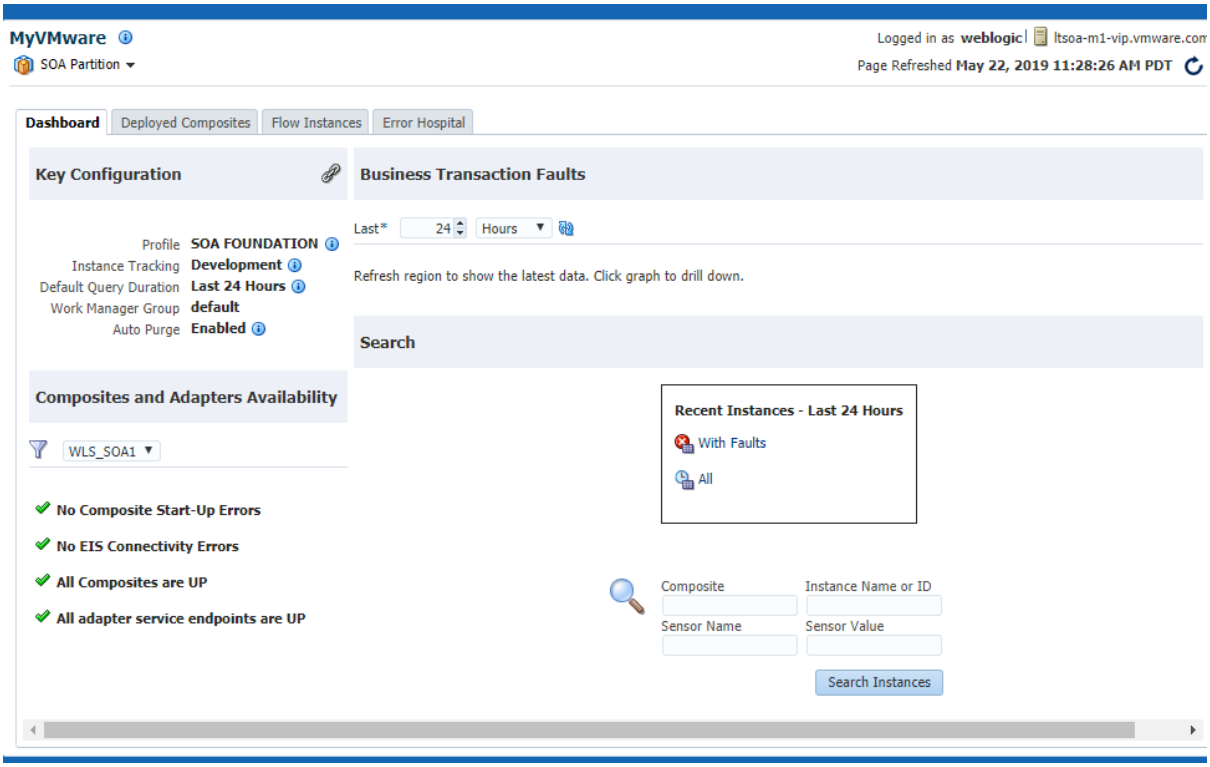


5.2 Monitoring Individual Portion of SOA

To Access overall status information for the individual partition.

1. Select Manage Partitions.
2. In the SOA partition column, select a specific partition.

    - Expand the SOA.
    - Select the soa-infra(WLS_SOA1).
    - Select the Specific Partition.



Avaya Engagement Development Platform Overview and Specification

6. Introduction – Engagement Development Platform

This document describes the high-level understanding of Avaya Engagement Development Platform features, functions, capacities, and limitations along with the tested product characteristics and capabilities, including feature descriptions, interoperability, performance specifications, security.

Avaya Engagement Development Platform provides a virtualized and secure application platform where Java programmers can develop and dynamically deploy advanced collaboration capabilities that extend the power of Avaya Aura®. A snap-in or service is the term used to describe this functionality. Customers, business partners, and Avaya developers can use the platform as the deployment vehicle for their applications or snap-ins.

Engagement Development Platform acts as the platform for many Avaya products such as the WebRTC Snap-in, Avaya Co-Browsing Snap-in, Avaya Real-Time Speech Snap-in, Engagement Designer, Context Store, and Work Assignment.

Engagement Development Platform provides the following benefits:

- Customers, partners, and Avaya organizations can rapidly develop snap-ins and applications that are deployed on the Engagement Development Platform.
- Developers can focus on building the collaboration snap-ins they need, without the need to develop a robust platform on which collaboration snap-ins are deployed and run.
- A robust Software Development Kit (SDK) with an easy-to-use API. Developers need not understand the details of call processing to develop new capabilities.
- The ability to perform operations such as:
    - Intercepting calls in to and out of the enterprise.
    - Redirecting calls to an alternate destination.
    - Blocking calls and optionally playing an announcement to the caller.
    - Changing the presented caller ID of the calling or called party.
- The ability to place an outbound call to play announcements and collect digits.
- The ability to invoke web services for added functionality. The ability to expose webpages and web services for invocation by remote browsers and applications.
- A Collaboration Bus that allows snap-ins to leverage each other's capabilities through point-to-point and publish/subscribe messaging patterns.
- A Common Data Manager framework that snap-ins use to access common information stored on System Manager.

7. Overview

Engagement Development Platform is a powerful snap-in delivery platform that provides Unified Communications and Contact Center customers and partners the ability to quickly deliver capabilities using the skill sets of today's enterprise and cloud application developers.

- Connector snap-ins provide access to email, Clickatell SMS (text messaging), and Scopia (conferencing) host applications.
- The ability to add or replace Trust and Identity Certificates for increased security.
- Tools that log and monitor operations and provide troubleshooting support.
- High availability.
- Third-party ability to create custom Connectors that provide access to their (external) application or service.
- Dynamic task types.

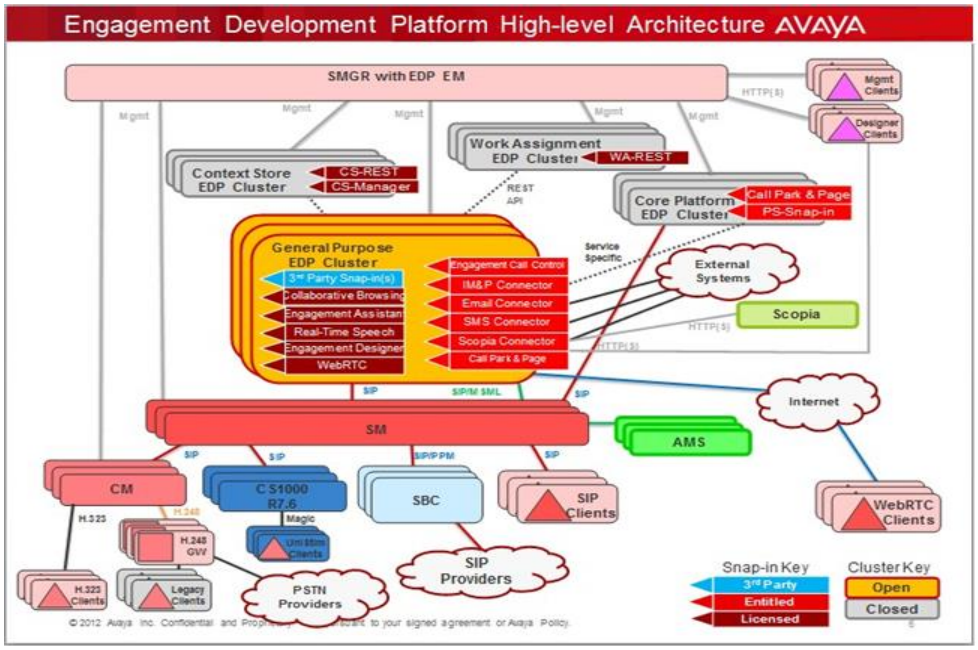
VMware Deployment

Engagement Development Platform is deployed into a VMware virtualized environment. It is delivered as a VMware vAppliance in Open Virtual Appliance (OVA) format and runs on a customer- provided VMware instance (standard edition or better).

Since the Engagement Development Platform is deployed in a virtualized VMware environment, all the snap-ins are deployed into the virtualized environment with no additional work needed on the part of the snap-in developer.

8. High-level Architecture

The following diagram provides a high-level illustration of the components of an Avaya Engagement Development Platform solution.



9. Features

Engagement Development Platform supports the following features:

Callable Services

Callable Services are the services whose features are invoked to originate or receive a call.

Real-Time Speech Improvements

Avaya Real-time Speech is an Avaya Engagement Development Platform snap-in that facilitates interactive speech searches on active voice calls. The snap-in provides a set of RESTful Web services that enable developers to incorporate real-time speech search capabilities in their solutions. To increase the efficiency, the Engagement Development Platform supports a mixed audio stream as the input to the speech search engine. This feature reduces the number of channels used on a call by up to 50%.

Engagement Development Platform also supports speech search by reference.

Trusted Host by cluster

The Engagement Development Platform Element Manager enables the administrator to configure the trusted host list on a per-cluster basis, such that only trusted hosts for that cluster can have http(s) access. The Engagement Development Platform Element Manager also enables the administrator to configure the HTTP(S) CORS host list on a cluster basis, such that only configured hosts for that cluster shall be allowed to access Cross-origin Resource Sharing.

10. Feature Description

10.1 Software Development Kit

Engagement Development Platform includes a Software Development Kit (SDK) for Java developers to create their collaboration snap-ins to run on the Engagement Development

Platform server. Any Java programmer can build, test, and deploy a custom snap-in. No specialized telecommunications expertise is needed.

The Engagement Development Platform SDK provides a rich set of developer collateral including code examples, video tutorials, online API documentation, and discussion forums.

10.2 Development Environment and Tools

Engagement Development Platform supports the following development environment and builds/ packaging tools:

- Any Java IDE can be used to develop snap-ins. Eclipse is the IDE used and recommended by the Avaya Engagement Development Platform team.
- The SDK includes Maven tools to build and package snap-ins. Although the use of Maven is not required to build and deploy snap-ins, it is the tool used and recommended by the Avaya Engagement Development Platform team. An Eclipse plug-in is available for the SDK.

10.3 Clusters

Clustering is the grouping of one or more Engagement Development Platform servers that can be managed together. An Engagement Development Platform cluster thus consists of one or more Engagement Development Platform servers. An Engagement Development Platform server belongs to only one cluster at a time.

An Engagement Development Platform server belongs to only one cluster at a time and has snap-ins installed. Snap-in installation is at the cluster level. This implies that all the servers in a single cluster will have the same snap-is.

10.4 Cluster Profiles

A cluster profile is a pre-loaded template that contains cluster attributes. The cluster profile specifies the fixed and variable attributes in a cluster. A set of cluster profiles are pre-loaded on the Element Manager. For every cluster profile, there is a set of required snap-ins. The required snap-ins are mandatory. Ensure that all the required snap-ins are loaded. Some cluster profiles may also have optional snap-ins. You can choose to install any of the optional snap-ins.

General Purpose Cluster Profile: A General-Purpose cluster is an open type cluster where you can install any type of snap-in or service. The minimum number of servers for a general-purpose cluster is 1. General Purpose Large cluster profile: An open cluster that mainly supports the Engagement Call Control solution.

Core Platform Cluster Profile: A closed cluster that supports up to 10 Engagement Development Platform servers. Install snap-ins like Presence Services and Call Park and Page Snap-in on this cluster.

Product-specific Cluster Profiles: Cluster profiles like Context Store profile or a Work Assignment profile are product specific. These cluster profiles have a specified list of required and optional snap-ins that you can install. If you attempt to install an unlisted snap-in for this cluster profile, the installation fails, and the system displays an error message.

Clustering Capabilities

Use the clustering functionality to:

- Create a new cluster and assign a cluster profile to a cluster.
- Edit clusters and cluster attributes.
- Delete clusters
- Add or remove servers from a cluster.
- Install or remove snap-ins across instances in a cluster.
- Manage resources for logging.
- Select product-specific cluster profiles like Context Store or Work Assignment.

Data Grid for Clusters

Engagement Development Platform supports data grid configuration on a cluster. The data grid is shared by all the servers in a cluster. If a server needs to find the data residing on another server on the cluster, a Lookup service is required. The Lookup service is hosted on two designated Lookup servers in a cluster.

10.5 Service Profiles

A Service Profile is an administered group of snap-ins. A snap-in is administered to have different snap-in attributes for each specific Service Profile. Therefore, the same snap-in can be tailored using the snap-in attributes to meet the needs of different users or groups.

11. Snap-in Types

Snap-ins that deploy on the Engagement Development Platform are categorized as follows. A given snap-in can fall into more than one category. The categories are not mutually exclusive.

Call Intercept Snap-ins – Called Party and Calling Party Snap-ins

All incoming and outgoing calls between the PSTN and the enterprise can take full advantage of Call Intercept snap-ins that run on the Engagement Development Platform. This is true regardless of the type of endpoint (H.323 or SIP) and the type of trunk (ISDN or SIP). station-to-station calls within the enterprise cannot invoke Call Intercept snap-ins even if the endpoints are SIP endpoints. There are two types of Call Intercept snap-ins:

- Based on who is being called, a Called Party snap-in looks at the configuration data for that called party to determine how to handle the call.
- Based on who is placing a call, a Calling Party snap-in looks at the configuration data for that calling party to determine how to handle the call.

11.1 Outbound Calling Snap-ins

Outbound Calling snap-ins initiate calls to phone numbers to play pre-recorded announcements and optionally detecting button presses from the called phone.

Outbound Calling snap-ins also initiate two-party calls to join two participants together in a call. The calling party is called first, and after the answer, a call is initiated to the called party. After the called party answers both the participants talk to each other. The Click to Call application is an example of a two-party Outbound Calling snap-in.

11.2 Collaboration Bus-invoked Snap-ins

The Collaboration Bus is a core module within the Engagement Development Platform that enables snap-ins to send messages to other snap-ins to leverage the functionality of the other snap-ins.

Collaboration Bus-invoked snap-ins perform some action when they receive a message from another snap-in on the Collaboration Bus.

The email connector snap-in is an example of a Collaboration Bus-invoked snap-in.

11.3 Connector Snap-ins

Engagement Development Platform includes several connector Snap-ins that provide access to external host applications. These built-in connector snap-ins communicate over the Collaboration Bus with snap-ins that request them. Connector snap-ins are available for:

- Email.
- Clickatell SMS (Text Messaging).
- Scopia (conferencing).
- Eventing Framework.

One-NDS 9 SP2

Installation Update Manual

12. Introduction – Installation Update Manual

One-NDS (Network Directory Server) is the central element in Subscriber Data Management (SDM) solution by Nokia, it is specifically designed to enable the use of a common centralized database by multiple applications through the support of open data access protocols. It is a real-time, resilient, and distributed data and application-hosting environment built to support 2G, 3G, and LTE networks. One-NDS not only consolidates subscriber profile data but enables rapid and cost-effective development of future applications and services.

One-NDS supports applications such as Nokia HSS/HLR, EIR, MNP, AAA, PCRF, or DDE, including 3rd party applications. All the front ends are deployed as data-less applications in record time and with less risk of errors because subscriber data is stored in a central repository with a single point of provisioning.

The Operating System (OS) is configured to meet the needs of operating One-NDS in the following areas:

- Redundant operation of network links.
- Disks are managed by logical volume management.
- Operation under demanding security conditions.

One-NDS is based on the '**in-memory' directory server**. The server roles in the One-NDS system are as follows:

- Install Server.
- DS Server (R-DSA, BE-DSA).
- PGW-DSA Server (includes Notification Manager).
- PGW Server.
- ADM Server (includes Status Service).

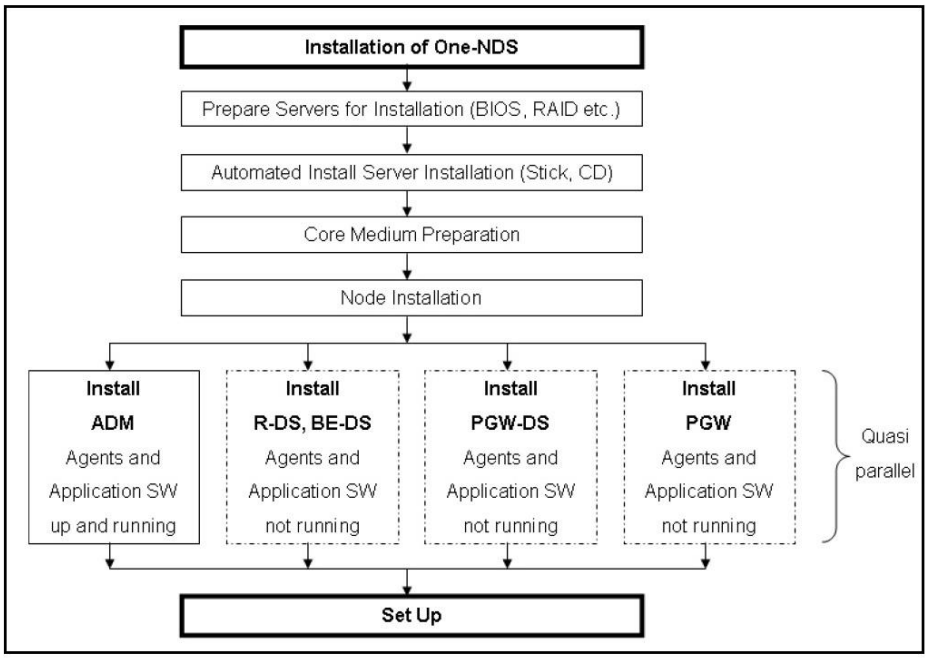
13. Installation Update

One-NDS 9 SP2 runs on *SuSE* Linux Enterprise Server operating system (SLES11).

The following are One NDS components:

- Directory Server (DS).
- Provisioning Gateway Directory Server (PGW-DS).
- Notification Manager (NTF).
- Install Server (INS).
- Provisioning Gateway Server (PGW).
- One-NDS Administrator (ADM).

Installation Phases of One- NDS



13.1 Prerequisites

The following are the prerequisites:

- One-NDS 9 base (OneNDS-XX-X-9.0.0.15_FULL) or One-NDS 9 SP1 (OneNDS-XX-X-9.0.1.16_FULL / OneNDS-XX-X-9.0.0.15_UPDATE_01-16) release must be installed.
- All hotfixes released for the currently installed software version that is marked as pre-update relevant must be installed on the system.
- BIOS and firmware versions on all nodes must be installed.

13.2 Restrictions

During the update phase, if the nodes are in a mixed-mode, it is not possible to update or add new extension packages. The enhanced password policy feature is enabled only after **extendedPasswordSyn_ADM.ldif** from additionalData is applied to all DSA.

13.3 Updating the ADM Application Software

During the update process, no operations using the One-NDS Administrator are allowed.

For the update of the ADM nodes, the following rules apply:

1. Turn off automatic synchronization on active ADM.
    a. Navigate to **ADM Administration -> ADM Synchronization**.
    b. Unselect the checkbox against **Enable automatic ADM synchronization**.
    c. Click **Save**.
2. Force the ADM Synchronization on active ADM.
    a. Navigate to ADM Administration -> ADM Synchronization.
    b. Click Start ADM Synchronization.
3. Perform the update of active ADM through SufDirector.
4. After the update of active ADM is completed, **[C]ommit /[F]allback** performs the update on all standby ADM nodes. A warning regarding “ADM Synchronization” will appear on each standby ADM node, for example:
5. Perform the update on all nomadic ADM nodes if they are included in the existing system. There are two possibilities regarding nomadic ADM nodes:
    a. If the warning regarding “ADM Synchronization” appears on nomadic ADM node, the node finishes in the state **[F]allback / [R]etry / [C]ontinue**.
    b. If the warning regarding “ADM Synchronization” does not appear on the nomadic ADM node, the node finishes in state **[C]ommit / [F]allback**.In both cases continue in the next step.
6. If all standby ADM nodes are in state **[F]allback / [R]etry / [C]ontinue**and if all nomadic ADM nodes are in state **[F]allback / [R]etry / [C]ontinue or [C]ommit / [F]allback**, force the ADM Synchronization on active ADM.
    a. Navigate to ADM Administration -> ADM Synchronization.
    b. Click Start ADM Synchronization.
7. Select choice [R]etry on each standby and relevant nomadic ADM nodes (Note: Some nomadic ADM nodes could already be in state [C]ommit / [F]allback). Post-installation checks are done again, and update must end in [C]ommit / [F]allback state.
8. Turn on automatic synchronization on active ADM.
    a. Navigate to ADM Administration -> ADM Synchronization.
    b. Select the checkbox against **Enable automatic ADM synchronization**.
    c. Click **Save**.

13.4 Updating the Directory Server Software (BE-DSA, R-DSA)

DS is updated only if it is not in the “Primary” state. This ensures that the DSA is always in operation.

For the update of the directory server software, the following rules apply for the update:

1. Update One or something else the BE server out of all in advance and wait for not about 24 hours before the other updates are started. This helps to verify the possibly erroneous behavior of the new software.
2. Start with the BE servers on one site fulfilling the preconditions. For more experienced users, all sites can be processed in parallel (the appropriate Install Server must be chosen accordingly).
    - As sdfrun, enter the command ndsSysInfoon every DS or get the DS status through the NetAct.
    - Status is seen through GUI ADM (for more information.
3. “Secondary Synchronized” servers are updated immediately and start the update of these servers.
4. Update the next site BE servers, which are ‘Secondary Synchronized’.
5. Initiate a site of previously updated servers to take over the primary role by using the ADM GUI.
6. The state change of DS is also be done by using nds command:
    a. as sdfrun, log in to the DS in status ‘Secondary Synchronized Primary Standby’
    b. Enter the command, nds changeoverthat switches the server roles ‘Primary’ <→ ‘Secondary Synchronized Primary Standby’.
7. Update all former ‘Primary’ now ‘Secondary Synchronized’ DS in parallel.
8. Repeat the same procedure for the directory servers within the R- DSA.

14.5 Updating the PGW-DSA Software

1. Update one directory server out of all in advance and wait about 24 hours before the other updates are started. This helps to verify the possible erroneous behavior of the new software.
2. Start with directory servers on one site fulfilling the preconditions (primary standby).
3. Secondary Synchronized” servers are updated immediately and start the update of these servers in parallel.
4. Initiate all previously updated servers to take over the primary role by using the ADM GUI.
5. Update all former ‘Primary’ now ‘Secondary Synchronized’ DS in parallel.

13.6 Updating the PGW Application Software

Update PGW one by one, not in parallel, to prevent provisioning outage.
Perform the following steps for each PGW:

1. ExP hotfixes must be uninstalled before update PGW according to release notes for ExP hotfixes.

   Reason is that PGW deletes any class files in its directories. Therefore, if any ExP hotfix delivered class files, they are lost with an update and the hotfix is no longer properly installed. Therefore, ExP causes issues that PGW cannot start.

2. Perform the update of PGW.

   Hint: Under certain conditions, the removal of PGW core hotfixes may fail during the update. In that case, perform a `SufDirector` fallback, remove the PGW core hotfixes by using the following script under `provgw` or root user and retry the update.

   ```
   # /etc/provgw/scripts/remove-hotfixes.sh
   ```

3. After the successful PGW software update, reinstall ExP hotfixes uninstalled in step 1.