Copyright Notices

Copyright © 2020 OneSpan North America Inc. All rights reserved.

Trademarks

OneSpan™, DIGIPASS® and CRONTO® are registered or unregistered trademarks of OneSpan North America Inc. and/or OneSpan International GmbH (collectively "OneSpan") in the U.S. and other countries.

OneSpan reserves all rights to the trademarks, service marks and logos of OneSpan and its subsidiaries.

All other trademarks or trade names are the property of their respective owners.

Intellectual Property

OneSpan Software, documents and related materials ("Materials") contain proprietary and confidential information. All title, rights and interest in OneSpan Software and Materials, updates and upgrades thereof, including software rights, copyrights, patent rights, industrial design rights, trade secret rights, sui generis database rights, and all other intellectual and industrial property rights, vest exclusively in OneSpan or its licensors. No OneSpan Software or Materials may be downloaded, copied, transferred, disclosed, reproduced, redistributed, or transmitted in any form or by any means, electronic, mechanical or otherwise, for any commercial or production purpose, except as otherwise marked or when expressly permitted by OneSpan in writing.

Disclaimer

OneSpan accepts no liability for the accuracy, completeness, or timeliness of content, or for the reliability of links to and content of external or third party websites or materials.

OneSpan shall have no liability under any circumstances for any loss, damage, or expense incurred by you, your company, or any third party arising from the use or inability to use OneSpan Software or Materials, or any third party material made available or downloadable. OneSpan will not be liable in relation to any loss/damage caused by modification of these Legal Notices or content.

Reservation

OneSpan reserves the right to modify these Notices and the content at any time. OneSpan likewise reserves the right to withdraw or revoke consent or otherwise prohibit use of the OneSpan Software or Materials if such use does not conform to the terms of any written agreement between OneSpan and you, or other applicable terms that OneSpan publishes from time to time.

Contact Us

Phone: 1-855-MYESIGN

e-Mail: support@onespan.com

Customer Support: https://www.esignlive.com/customer-support

Resource Center: https://www.esignlive.com/resource-center

Company Website: https://www.onespan.com

Date: July 6, 2020
CONTENTS

PART I: INTRODUCTION ........................................ 1

Chapter 1: About This Document ...................... 2

- Document Purpose ............................................. 2
- Document Audience .......................................... 2
- The 7.2 Release .................................................. 2
  - Architecture .................................................. 3
  - Deployment Scenarios ....................................... 3
  - Benefits for On-Premises Customers ......................... 3
  - The Integration Model ....................................... 4
  - Required Installation Time .................................. 4
  - Components .................................................... 4
  - Product Documentation ..................................... 7
- Document Organization ....................................... 9
- Document Conventions ....................................... 9

Chapter 2: Deployment Overview ...................... 11

- Deployment Infrastructure ................................... 11
- Hardware Requirements ...................................... 13
  - Minimum Requirements ..................................... 13
- Software Requirements ...................................... 13
- User Access Requirements ................................... 23
- Tenants .......................................................... 24
- Platform Configuration ...................................... 25
  - Resource Manager .......................................... 25
  - Admin Console .............................................. 26
  - Client Components .......................................... 26
- Deployment Topology ........................................ 27
- Deployment Road Map ....................................... 28
- Packaging ........................................................ 33

PART II: THE INSTALLER ................................. 34

Chapter 3: Using the Installer ......................... 35

- Installing Components ....................................... 36
- Uninstalling Components .................................... 51

PART III: REQUIRED PLATFORM SOFTWARE ....... 53

Chapter 4: Deploying the Audit Service ............ 54

- Installing the Audit Service ................................ 54
- Uninstalling the Audit Service ............................. 56

Chapter 5: Deploying the User Manager ............. 58

- Installing the User Manager ................................ 58
- Uninstalling the User Manager ............................. 61

Chapter 6: Deploying the Resource Manager ........ 63

- Installing the Resource Manager ......................... 63
- Uninstalling the Resource Manager ....................... 64

Chapter 7: Deploying the Admin Console .......... 65

- Installing the Admin Console .............................. 65
- Uninstalling the Admin Console ........................... 69
## Chapter 8: Deploying the Core Component

- Installing the Core Component .................................................. 71
  - General Prerequisites .............................................................. 71
  - Installing the Core Component on WebSphere .......................... 72
  - Installing the Core Component on JBoss or Wildfly .................. 79
  - Installing the Core Component on WebLogic ............................ 81
- Uninstalling the Core Component .............................................. 84
  - Uninstalling the Core Component on WebSphere ....................... 85
  - Uninstalling the Core Component on JBoss or Wildfly ............... 86
  - Uninstalling the Core Component on WebLogic ........................ 87

## Chapter 9: Deploying the Event Manager

- Installing the Event Manager .................................................... 89
  - Installing the Event Manager on WebSphere ............................ 89
  - Installing the Event Manager on JBoss or Wildfly ................... 91
  - Installing the Event Manager on WebLogic .............................. 91
- Uninstalling the Event Manager ............................................... 92
  - Uninstalling the Event Manager on WebSphere ........................ 92
  - Uninstalling the Event Manager on JBoss or Wildfly ............... 93
  - Uninstalling the Event Manager on WebLogic ........................ 93

## Chapter 10: Deploying the e-Mail Notification Manager

- Installing the Scheduler ......................................................... 94
  - Installing the Scheduler on WebSphere ................................. 94
  - Installing the Scheduler on JBoss or Wildfly .......................... 98
  - Installing the Scheduler on WebLogic .................................... 100
- Installing the Notifier ............................................................ 102
  - Installing the Notifier on WebSphere .................................... 102
  - Installing the Notifier on JBoss or Wildfly ............................ 104
  - Installing the Notifier on WebLogic ...................................... 105
- Uninstalling the Scheduler ..................................................... 107
  - Uninstalling the Scheduler on WebSphere .............................. 107
  - Uninstalling the Scheduler on JBoss or Wildfly ...................... 108
  - Uninstalling the Scheduler on WebLogic ............................... 109
- Uninstalling the Notifier ....................................................... 109
  - Uninstalling the Notifier on WebSphere ............................... 110
  - Uninstalling the Notifier on JBoss or Wildfly ....................... 110
  - Uninstalling the Notifier on WebLogic ................................. 110

## Chapter 11: Deploying PDF Document Engines

- Using Multiple PDF Document Engines .................................... 112
- Tenant and Non-Tenant Environments ...................................... 112
- Installing PDF Document Engines ......................................... 113
  - Installing a New PDF Document Engine on Linux ................... 114
  - Installing a New Doc Engine on Windows with Apache ............. 123
  - Installing a New Doc Engine on Windows with IIS ................... 126
  - Upgrading a PDF Document Engine on Linux ............................ 130
  - Upgrading a PDF Document Engine on Windows ....................... 130
  - Configuring a PDF Document Engine ..................................... 131
  - Configuring the Trusted CA Folder for HTTPS ......................... 156
  - Enabling SOAP Security ....................................................... 157
- Uninstalling PDF Document Engines ........................................ 157

## PART IV: OPTIONAL PLATFORM SOFTWARE

## Chapter 12: Deploying e-Witness

- Installing e-Witness ............................................................... 161
- General Prerequisites ........................................................... 161
- Installing e-Witness on WebSphere ......................................... 164
CONTENTS

Preparing an Environment for WebSphere ......................... 312
Preparing an Environment for JBoss 6 or Wildfly ................. 314
  Startup Modes ............................................. 314
  Server Profiles .......................................... 315
  Prerequisites .............................................. 315
Preparing an Environment for WebLogic ......................... 316
Creating a JDBC Provider .................................... 316
  Creating a JDBC Provider for WebSphere ..................... 316
  Creating a JDBC Provider for JBoss 6 or Wildfly .......... 317
  Creating a JDBC Provider for WebLogic ..................... 318

Appendix I: Configuring IIS & Optimizing Performance319

Appendix J: Running the DE as an Application User .. 322
  Starting the Doc Engine on RHEL/CentOS 6 as an Application User .... 322
  Starting the Doc Engine on RHEL/CentOS 7 as an Application User .... 323
  Uninstalling the Doc Engine as an Application User ............. 325

Appendix K: Glossary ................................. 327
## FIGURES

<table>
<thead>
<tr>
<th>Figure Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical deployment infrastructure</td>
<td>12</td>
</tr>
<tr>
<td>Platform configuration overview</td>
<td>25</td>
</tr>
<tr>
<td>Deployment topology</td>
<td>28</td>
</tr>
<tr>
<td>Deployment Road Map</td>
<td>30</td>
</tr>
<tr>
<td>Deployment strategy for WebSphere</td>
<td>31</td>
</tr>
<tr>
<td>Deployment strategy for JBoss or Wildfly</td>
<td>32</td>
</tr>
</tbody>
</table>
Table 1-1: Software components provided by OneSpan .......................... 5
Table 1-2: Guides for OneSpan Sign ....................................................... 7
Table 1-3: Font conventions ............................................................... 10
Table 1-4: Icon conventions ............................................................... 10
Table 2-1: Recommended minimum server-side hardware requirements ........ 13
Table 2-2: Supported third-party products for the Core Component, etc. ....... 15
Table 2-3: Supported (OS, App. Server) & (OS, DB Server) combinations ....... 17
Table 2-4: Additional supported third-party products ................................. 17
Table 3-1: Parameters in the configuration file "EnvConfig.xml" ................. 42
Table 3-2: Modules that can be installed/uninstalled ................................. 49
Table 3-3: Mapping between Installer names & databases ........................... 50
Table 4-1: Audit Service Data Source parameters .................................... 54
Table 4-2: Scheduler Queue parameters ................................................ 97
Table 5-1: User Manager’s Data Source settings ....................................... 59
Table 7-1: OneSpan Sign Data Source parameters .................................... 67
Table 8-1: Error Queue parameters ....................................................... 76
Table 8-2: Post-Online Queue parameters ............................................... 76
Table 8-3: Concatenate Documents Queue parameters ................................ 77
Table 8-4: Notifier Response Queue parameters ....................................... 77
Table 8-5: Data Source parameters for the Document Retriever Pool DB ....... 78
Table 9-1: Event Manager Queue parameters .......................................... 90
Table 10-1: e-Mail Notification Manager’s Data Source settings .................. 97
Table 10-2: Scheduler Queue parameters ............................................... 97
Table 10-3: Notifier Queue parameters .................................................. 104
Table 11-1: Environment variables for the PDF Document Engine ............... 122
Table 11-2: Configuration parameters for the PDF Document Engine .......... 137
Table 12-1: PrintToPdf Queue parameters ............................................. 166
Table 13-1: Files in the Archival Module’s home directory ......................... 175
Table 15-1: OneSpan Application & URL Mapper Data Source parameters ...... 188
Table 15-2: Statistics Queue parameters ................................................. 189
Table 15-3: Expire Package Queue parameters ........................................ 190
Table 15-4: Package Callback Queue parameters .................................... 190
Table 15-5: Connector Framework Callback Queue parameters .................. 190
Table 15-6: Salesforce Package Event Queue parameters ............................ 191
Table 15-7: Salesforce Upload Completed Queue parameters ...................... 191
Table 15-8: Evidence Summary Queue parameters .................................... 192
Table 15-9: Package Reminders Queue parameters .................................... 192
Table 15-10: Email Queue parameters ................................................... 193
Table 15-11: SMS Queue parameters .................................................... 193
Table 15-12: Async Business Queue parameters ...................................... 193
Table 15-13: Environment variables for the Classic User Experience .......... 199
Table 15-14: Environment variables for the New User Experience .............. 209
Table 15-15: Parameters in settings.json .............................................. 221
Table 15-16: Parameters in asseco-settings.json ..................................... 223
Table 15-17: Parameters in validatedID-settings.json ............................... 224
Table 15-18: Error Codes ................................................................. 225
Table 16-1: Configuration parameters required for a Proxy Server .............. 237
TABLES

Table 18-1: Configuration parameters required by the Application. . . . . . . . . . 268
Table 18-2: Optional configuration parameters for the Application. . . . . . 273
Table 18-3: Configuration parameters in "saml.properties". . . . . . . . 276
PART I: INTRODUCTION

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: About This Document</td>
<td>2</td>
</tr>
<tr>
<td>2: Deployment Overview</td>
<td>11</td>
</tr>
</tbody>
</table>
CHAPTER 1: About This Document

This chapter provides an overview of this document by discussing:

- **Document Purpose (page 2)**
- **Document Audience (page 2)**
- **The 7.2 Release (page 2)**
- **Document Organization (page 9)**
- **Document Conventions (page 9)**

**Document Purpose**

The electronic-signature solution called OneSpan Sign provides a complete E-Signature Process Management platform for the Web, including preparing, distributing, reviewing, signing, and downloading documents.

This guide describes how to perform an on-premises deployment of OneSpan Sign.

Each installation procedure describes a first-time installation, unless it explicitly states that it describes an upgrade.

**Document Audience**

This guide’s intended audience is the personnel responsible for performing an on-premises deployment of OneSpan Sign. They must have:

- Expertise with Java technologies
- Expertise configuring the type of supported Application Server where OneSpan Sign will be deployed (see Table 2-2)
- Experience creating databases
- General networking and security knowledge
- Suitable permissions for all relevant servers
- System Administration (Linux, Windows)

**The 7.2 Release**

This section describes the following aspects of the 7.2 release of OneSpan Sign:

- **Architecture (page 3)**
- **Deployment Scenarios (page 3)**
- **Benefits for On-Premises Customers (page 3)**
- **The Integration Model (page 4)**
- **Required Installation Time (page 4)**
- **Components (page 4)**
- **Product Documentation (page 7)**
The product has two main functional sectors:

- **OneSpan Sign Platform** (aka *The Platform*) — This is the product’s core transactional and business logic.
- **OneSpan Sign Application** (aka *The Application*) — This is the product’s front end. Users, administrators, and integrators can interact with the Application using its GUI, its API, its SDKs, or its connectors. The Application is built on our REST Integration Model (for more on that model, see the *Integrator’s Guide*).

### Deployment Scenarios

*OneSpan Sign* customers can choose either of the following deployment scenarios:

- **SaaS** (*Software as a Service*) — The product is centrally hosted on the public Cloud. Customers access the product by subscribing to an online service.
- **On-premises** — The product is installed on the customer’s premises, usually behind a firewall.

All users — SaaS and on-premises — interact with the product via the Application.

*OneSpan Sign* 6.0 unified these two options in the sense that, for the first time, both scenarios provided essentially the same functionality from the same code base.

Despite their unification, a few minor differences persist between those two deployment scenarios. In particular, the following features are currently available only in SaaS deployments:

- A GUI footer with links to the *OneSpan Sign Trust Center*, terms and conditions, and product documentation
- The ability to review account-subscription details and quotas using an *Administration* tab
- eSignLive for Box

### Benefits for On-Premises Customers

The Application became available to on-premises customers in *OneSpan Sign* 6.0. It offers on-premises customers the following major benefits:

- All customers can use a rich GUI that is out-of-the-box. On-premises customers no longer have to develop a GUI from scratch, so the expense and time required for an on-premises deployment is much reduced.
- A Sender GUI enables users to manually create and manage e-signature transactions in a sender-driven (*ad hoc*) manner. Previously, on-premises customers could create and manage transactions only via a software-integration layer (i.e., "straight-through processing" that is...
The Integration Model

OneSpan Sign supports a single REST Integration Model that uses a REST API, a Java SDK, and a .Net SDK. For more on this model, see the Integrator’s Guide.

The Application’s REST API and SDK are backwards-compatible. This means, for example, that if you coded against version 11.1 of the SaaS REST API and then deployed OneSpan Sign 7.2, your custom integration will still work in 7.2.

Before the 6.5 release of OneSpan Sign, the product supported an XSLT Integration Model for on-premises deployments. That model was based on XSLT workflow templates and a SOAP API. It is no longer supported.

Required Installation Time

OneSpan Sign offers tremendous capability and flexibility, but it’s crucial that it be installed properly.

The product’s installation should take as little as one day if:

• It is conducted by personnel with the required expertise; see Document Audience (page 2).
• The required environment has already been prepared; see Chapter 2 and Preparing an App Server Environment (page 312).

Components

IMPORTANT: The OneSpan Sign components you can deploy and use are defined and governed by your license agreement and the corresponding software license file as provided by OneSpan.

The software components in Table 1-1 are provided by OneSpan. Before OneSpan Sign 6.0, all Platform components were available only via an on-premises deployment, and all Application components were available only via a SaaS deployment.
### Table 1-1: Software components provided by OneSpan

<table>
<thead>
<tr>
<th>COMPONENT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Required Platform Components</strong></td>
<td></td>
</tr>
<tr>
<td>Core Component</td>
<td>The primary component that supplies the Platform’s core functionality (e.g., coordinate transactions and events, access the Data Model). It runs on an Application Server.</td>
</tr>
<tr>
<td>PDF Document Engine</td>
<td>A component that performs all operations that manipulate a PDF document (signing, etc.)</td>
</tr>
<tr>
<td>e-Mail Notification Manager™</td>
<td>A component that sends e-Mail invitations and reminders to sign documents</td>
</tr>
<tr>
<td>Administration Console</td>
<td>A module that enables OneSpan Sign Administrators to configure and maintain the OneSpan Sign Platform</td>
</tr>
<tr>
<td>Resource Manager</td>
<td>A module that manages resources (e.g., templates, properties, Data Sources)</td>
</tr>
<tr>
<td>User Manager</td>
<td>A module that manages user credentials and permissions</td>
</tr>
<tr>
<td>Event Manager</td>
<td>A module that monitors internal events</td>
</tr>
<tr>
<td>Audit Service</td>
<td>A service that can create reports about the past activities of OneSpan Sign users and Admin Console users</td>
</tr>
<tr>
<td><strong>Optional Platform Components</strong></td>
<td></td>
</tr>
<tr>
<td>e-Witness™</td>
<td>A module that enables authorized parties to find, review, and verify a stored e-signature process. It also provides a utility to securely export Electronic Evidence™ to a PDF file and/or a Web Archive HTML file. Before OneSpan Sign 5.1, e-Witness was known as the ProcessReviewer.</td>
</tr>
<tr>
<td>Archival Module™</td>
<td>A component that enables administrators to archive data from and restore data to the OneSpan Sign Core Database</td>
</tr>
<tr>
<td>e-Vault Manager™</td>
<td>A companion server product that can create, control, and transfer ownership of financial instruments called Transferable Records (e.g., e-Notes in a mortgage process).</td>
</tr>
<tr>
<td><strong>Required Application Components</strong></td>
<td></td>
</tr>
<tr>
<td>OneSpan Sign Application Backend</td>
<td>Back-end support for the REST Integration Model (REST API, Java &amp; .Net SDKs) and the OneSpan Sign Application</td>
</tr>
<tr>
<td>OneSpan Sign BackOffice</td>
<td>A module that enables OneSpan Sign Administrators to configure and maintain the OneSpan Sign Application</td>
</tr>
<tr>
<td>BackOffice Support</td>
<td>A service module that supports the internal functioning of OneSpan Sign BackOffice</td>
</tr>
<tr>
<td>OneSpan Sign Scheduler</td>
<td>An Application component that schedules a notification to be sent at a future time</td>
</tr>
<tr>
<td>OneSpan Sign Authentication</td>
<td>An Application tool that authenticates user accounts</td>
</tr>
</tbody>
</table>
### COMPONENT DESCRIPTION

<table>
<thead>
<tr>
<th>COMPONENT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction Status API</td>
<td>An API that enables integrators to build an external GUI to monitor and manage OneSpan Sign transactions</td>
</tr>
<tr>
<td><strong>Optional Application Components</strong></td>
<td></td>
</tr>
<tr>
<td>OneSpan Sign Application Frontend</td>
<td>The Application’s User Experience (UX). In fact, OneSpan Sign offers two UXs:</td>
</tr>
<tr>
<td></td>
<td>• The <em>Classic User Experience</em> for signers, senders and administrators. This UX became available in OneSpan Sign 6.0, and is still available.</td>
</tr>
<tr>
<td></td>
<td>• The <em>New User Experience</em> for senders and administrators. For sender and administrator actions, this UX represents an enhancement over the Classic UX. It became available in OneSpan Sign 6.1.</td>
</tr>
<tr>
<td>OneSpan Sign Document Converter</td>
<td>An Application module that converts a document from DOC or ODT format to PDF format</td>
</tr>
<tr>
<td>Personal Certificate Client</td>
<td>An Application module used to sign documents with a personal digital certificate</td>
</tr>
<tr>
<td>OneSpan Sign Single Sign-On</td>
<td>An Application tool that authenticates user accounts against a supported external Authorization Server/Identity Provider</td>
</tr>
<tr>
<td>OneSpan Sign Mobile SDK</td>
<td>An Application feature that enables users to run e-signature processes from mobile devices</td>
</tr>
<tr>
<td>OneSpan Sign Equifax</td>
<td>An Application tool that authenticates user accounts created by OneSpan Sign BackOffice using Equifax</td>
</tr>
<tr>
<td>OneSpan Sign e-Notary Validation</td>
<td>An Application tool that potentially enables the online validation of notaries</td>
</tr>
<tr>
<td>OneSpan Sign URL Mapper</td>
<td>An Application tool that maps shortened URLs to complete Application URLs</td>
</tr>
<tr>
<td>OneSpan Sign Controller</td>
<td>An Application component that enables communication between the Platform and the Application</td>
</tr>
<tr>
<td>Connector Framework</td>
<td>A standalone Application component that enables individual connectors to connect to OneSpan Sign</td>
</tr>
<tr>
<td>OneSpan Sign for Microsoft SharePoint</td>
<td>An Application connector that enables OneSpan Sign users to obtain securely signed Microsoft SharePoint documents</td>
</tr>
<tr>
<td>OneSpan Sign for Salesforce</td>
<td>An Application connector that enables OneSpan Sign users to obtain securely signed Salesforce documents</td>
</tr>
<tr>
<td>eSignLive for Microsoft Dynamics CRM</td>
<td>An Application connector that enables OneSpan Sign users to obtain securely signed Microsoft Dynamics CRM documents. The name of this component is scheduled to become OneSpan Sign for Microsoft Dynamics CRM.</td>
</tr>
</tbody>
</table>
Table 1-2 lists the product documentation included in a release of OneSpan Sign.

### Table 1-2: Guides for OneSpan Sign

<table>
<thead>
<tr>
<th>DOCUMENT</th>
<th>PURPOSE</th>
<th>AUDIENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Guides for On-Premises Only</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Getting Started</td>
<td>Provides a brief introduction to OneSpan Sign</td>
<td>Anyone who wants a brief introduction to OneSpan Sign</td>
</tr>
<tr>
<td>On-Premises Deployment Guide (this document)</td>
<td>Describes how to deploy OneSpan Sign on premises</td>
<td>System Administrators, Database Administrators</td>
</tr>
<tr>
<td>Platform Administrator’s Guide</td>
<td>Describes how to configure and maintain the OneSpan Sign Platform (using the Administration Console)</td>
<td>OneSpan Sign Administrators</td>
</tr>
<tr>
<td>e-Witness User’s Guide</td>
<td>Describes how to find, review, and verify stored e-signature processes</td>
<td>Compliance personnel</td>
</tr>
<tr>
<td>Archival Module User’s Guide</td>
<td>Describes how to archive data from and restore data to OneSpan Sign’s Core Database</td>
<td>System Administrators, Database Administrators, OneSpan Sign Administrators</td>
</tr>
<tr>
<td>Supported Platforms</td>
<td>Summarizes the software requirements for OneSpan Sign and the e-Vault Manager</td>
<td>System Administrators, Database Administrators</td>
</tr>
<tr>
<td>Transaction Status API Deployment Guide</td>
<td>Describes how to deploy the Transaction Status API on an Application Server</td>
<td>System Administrators, Database Administrators</td>
</tr>
<tr>
<td>DOCUMENT</td>
<td>PURPOSE</td>
<td>AUDIENCE</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td><strong>Transaction Status API Integrator's Guide</strong></td>
<td>Describes how to integrate a third-party Web application with the Transaction Status API</td>
<td>Application Developers</td>
</tr>
<tr>
<td><strong>Guides for On-Premises &amp; SaaS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Integrator’s Guide</strong></td>
<td>Describes how to integrate a third-party Web application with the OneSpan Sign Application using the REST Integration Model</td>
<td>Application Developers</td>
</tr>
<tr>
<td><strong>Connector Guides</strong></td>
<td>Describes OneSpan Sign connectors to the following applications: (1) Salesforce; (2) Microsoft SharePoint; (3) Microsoft Dynamics CRM; (4) Box; (5) eSignLive Print Driver</td>
<td>Administrators (deployment), OneSpan Sign Application users (use), Application Developers (integration)</td>
</tr>
<tr>
<td><strong>Mobile Guides</strong></td>
<td>Describes various ways of running e-signature processes on mobile devices</td>
<td>Signers, senders, and Application Developers</td>
</tr>
<tr>
<td><strong>New Application User's Guide</strong></td>
<td>Describes the New User Experience for senders and administrators. For sender and administrator actions, this experience represents an enhancement over the Classic User Experience. It became available in OneSpan Sign 6.1.</td>
<td>Senders and administrators of the OneSpan Sign Application</td>
</tr>
<tr>
<td><strong>Classic Application User's Guide</strong></td>
<td>Describes the Classic User Experience for signers, senders and administrators. This experience became available in OneSpan Sign 6.0, and is still available.</td>
<td>Signers, senders, and administrators of the OneSpan Sign Application</td>
</tr>
<tr>
<td><strong>Application Administrator's Guide</strong></td>
<td>Describes how to configure and maintain the OneSpan Sign Application (using OneSpan Sign BackOffice)</td>
<td>Administrators of the OneSpan Sign Application</td>
</tr>
<tr>
<td><strong>SAML Admin Guide</strong></td>
<td>Describes how to enable members of OneSpan Sign accounts to log in to OneSpan Sign using &quot;Single Sign-On&quot; via SAML (Security Assertion Markup Language)</td>
<td>Administrators</td>
</tr>
</tbody>
</table>

© 2020 - OneSpan North America Inc.  DG-OSS-72 - All rights reserved. 8
This guide is divided into the following parts:

1. **PART I: INTRODUCTION** — Chapter 1 introduces the guide as well as *OneSpan Sign*. Chapter 2 provides a deployment overview.

2. **PART II: THE INSTALLER** — Chapter 3 describes how to use an Installer to deploy many *OneSpan Sign* modules in an automated manner.

3. **PART III: REQUIRED PLATFORM SOFTWARE** — These chapters describe how to manually deploy all *required* Platform modules.

4. **PART IV: OPTIONAL PLATFORM SOFTWARE** — These chapters describe how to manually deploy *optional* Platform modules.

5. **PART V: APPLICATION COMPONENTS** — This chapter describes how to manually deploy all Application modules.

6. **PART VI: OPTIONAL HARDWARE** — This chapter describes how to manually deploy optional hardware modules.

7. **PART VII: CONFIGURING THE SYSTEM** — These chapters describe how to configure the Platform and the Application.

8. **PART VIII: APPENDICES** — The appendices support the content in the guide’s chapters.

If you are installing components manually: (1) you must install all required Platform components (PART III) before any other components; (2) we **strongly recommend** that you follow the *Deployment Road Map* in Chapter 2.

**Document Conventions**

This guide uses the font conventions in Table 1-3 and the icon conventions in...
Table 1-3: Font conventions

<table>
<thead>
<tr>
<th>CONVENTION</th>
<th>USE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bold</strong></td>
<td>Highlights commands, menu names and choices, control names, entry-field names, icon names, and the first line of each procedure (e.g., &quot;click OK&quot;)</td>
</tr>
<tr>
<td><strong>Underlined</strong></td>
<td>Highlights cross-references and Web links (e.g., &quot;see Chapter 2&quot;). You can jump to links and cross-references by clicking the underlined text.</td>
</tr>
<tr>
<td><strong>Italics</strong></td>
<td>Highlights window names, book titles, new terms, and important text (e.g., &quot;the New Database window closes&quot;)</td>
</tr>
<tr>
<td><strong>Courier New</strong></td>
<td>Highlights file names, directory names, code examples, entered data, and message text (e.g., &quot;the script called awsng_sqlserver.sql&quot;)</td>
</tr>
</tbody>
</table>

Table 1-4: Icon conventions

<table>
<thead>
<tr>
<th>ICON</th>
<th>USE</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Notes" /></td>
<td>Notes</td>
</tr>
<tr>
<td><img src="image" alt="Warnings" /></td>
<td>Warnings</td>
</tr>
<tr>
<td><img src="image" alt="Tips or recommendations" /></td>
<td>Tips or recommendations</td>
</tr>
</tbody>
</table>

⚠️ You should implement this guide’s procedures in the order in which they appear (unless otherwise stated). If you fail to do so, the software may not work properly.

⚠️ The screenshots in this guide are illustrative only. They may differ from what you see when you deploy OneSpan Sign.
CHAPTER 2: Deployment Overview

This chapter provides an overview of the deployment by discussing:

- Deployment Infrastructure (page 11)
- Hardware Requirements (page 13)
- Software Requirements (page 13)
- User Access Requirements (page 23)
- Tenants (page 24)
- Platform Configuration (page 25)
- Deployment Topology (page 27)
- Deployment Road Map (page 28)
- Packaging (page 33)

Deployment Infrastructure

Figure 2-1 shows the infrastructure of a typical deployment. The OneSpan logo indicates infrastructure components with installed OneSpan Sign software.
Figure 2-1: Typical deployment infrastructure
Hardware Requirements

Minimum Requirements

Table 2-1 specifies the recommended minimum hardware requirements for the server-side machines in a OneSpan Sign production environment.

Table 2-1: Recommended minimum server-side hardware requirements

<table>
<thead>
<tr>
<th>MACHINE</th>
<th>RECOMMENDED MINIMUM HARDWARE</th>
</tr>
</thead>
</table>
| Application Server | • Intel or AMD 2.0 GHz  
• 30 GB of disk storage  
• 8 CPUs per Virtual Machine / Node hosting both Platform and Application JVMs (see Figure 2-3)  
• JVM memory (see Figure 2-3):  
  — 4 GB RAM for each Platform JVM and each Application JVM  
  — 2 GB RAM for the Administration JVM |
| PDF Document Engines | • Intel or AMD 2.0 GHz  
• 25 GB of disk storage (+ 4 GB temp space)  
• 4 GB RAM  
• 4 CPUs |
| Database Server | Consult your database vendor. |
| Machine for the OneSpan Sign Application Frontend | • Intel or AMD 2.0 GHz  
• 25 GB of disk storage (+ 2 GB temp space)  
• 2 GB RAM |
| Machine for the OneSpan Sign Document Converter | • Intel or AMD 2.0 GHz  
• 15 GB of disk storage  
• 4 GB RAM |

The actual hardware requirements depend on the number of documents, size of documents, number of transactions, and peak loads, among other factors. For example, the memory required by the Application Server and PDF Document Engines depends on the documents’ size and complexity.

The minimum hardware requirements for the client-side machine required by OneSpan Sign are those recommended for your client-side Operating System by its manufacturer.

Software Requirements

OneSpan Sign is supported on the third-party software specified in the next three tables. Specifically:

• Table 2-2 lists Operating Systems, Application Servers, and Database Servers that can support:
  • The following Platform components: Audit Service, User Manager,
Resource Manager, Core Component, Event Manager, e-Mail Notification Manager, e-Witness, Archival Module, and Admin Console

- The following Application components: OneSpan Sign Application Backend, OneSpan Sign BackOffice, OneSpan Sign Scheduler, OneSpan Sign Authentication, OneSpan Sign Single Sign-On, OneSpan Sign Mobile SDK, OneSpan Sign Equifax, OneSpan Sign e-Notary Validation, OneSpan Sign Controller, OneSpan Sign URL Mapper, and Transaction Status API

- Table 2-3 displays various (Operating System, Application Server) and (Operating System, Database Server) combinations. A check mark (✓) indicates that the corresponding combination is supported, subject to the limitations and constraints imposed by the relevant software providers. A dash (—) indicates that the combination is not supported.

- Table 2-4 specifies additional supported third-party software. All browsers in this table should be JavaScript-enabled and cookies-enabled.

Table 2-2 documents multiple supported platforms. When you copy software provided by OneSpan to a local File System, you must ensure that the target directory is named appropriately for your platform. For example, on Windows you can copy to a directory name with blank spaces, but on Linux you can copy only to a directory name with no blank spaces (e.g., Doc Engine vs Doc_Engine).

The terms for platform support are governed by the Silanis Software License and Maintenance Agreement (SLMA), and are described in the Platform Support Policy document provided by OneSpan Sign Support.

Unless otherwise stated, all software version numbers in Table 2-2 and Table 2-4 implicitly include all associated minor releases. For example, "WebSphere 8.0" means "WebSphere 8.0.x" where x is a variable (e.g., WebSphere 8.0.0.4).

The clocks of all OneSpan Sign components must be synchronized. This even includes components on different nodes in a cluster. The recommended time zone is UTC (Coordinated Universal Time).

Table 2-4 contains multiple references to Internet Explorer, but none of them is intended to include the Mobile version of Internet Explorer. That version is not supported.

OneSpan Sign uses standard JNDI Data Sources, and drivers configured for the Application Server. For transaction management, they use the JTA (Java Transaction API) implemented by the Application Server.
### Table 2-2: Supported third-party products for the Core Component, etc.

<table>
<thead>
<tr>
<th>ITEM</th>
<th>SUPPORTED ON</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operating System</strong></td>
<td></td>
</tr>
<tr>
<td>Red Hat Enterprise Linux Server</td>
<td>• 7.x, 6.x</td>
</tr>
<tr>
<td>CentOS</td>
<td>• 7.x, 6.x</td>
</tr>
<tr>
<td>Microsoft Windows Server</td>
<td>• 2016 Standard or Datacenter</td>
</tr>
<tr>
<td></td>
<td>• 2012 Standard or Datacenter</td>
</tr>
<tr>
<td></td>
<td>• 2012 R2 Standard or Datacenter</td>
</tr>
<tr>
<td>ITEM</td>
<td>SUPPORTED ON</td>
</tr>
<tr>
<td>------</td>
<td>--------------</td>
</tr>
<tr>
<td><strong>IBM WebSphere Application Server Network Deployment</strong>&lt;sup&gt;1, 2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>• 9.0 (Fix Pack 11) with the IBM Java SDK 8.0</td>
<td></td>
</tr>
<tr>
<td>• 8.5.5 (Fix Pack 9 or higher)&lt;sup&gt;3, 4&lt;/sup&gt; with the IBM Java SDK 8.0</td>
<td></td>
</tr>
<tr>
<td>• Optional: IBM WebSphere MQ 7.0, 8.0.0.4, or 9.0.3 as an external messaging provider (only these 3 versions are certified for clusters).</td>
<td></td>
</tr>
</tbody>
</table>

<sup>1</sup>If your WebSphere server is configured to use *Lightweight Third-Party Authentication* (LTPA), you must disable the server's "application security".

<sup>2</sup>The Java Cryptography Extension (JCE) *Unlimited Strength Jurisdiction Policy Files* must be installed for your version of the Java SDK.

<sup>3</sup>If you are using IBM HTTP Server as a Load Balancer, IBM HTTP Server needs the same Fix Pack as the App. Server (e.g., Fix Pack 9). If instead your Load Balancer (e.g., Apache) is using the WebSphere Plugin, the WebSphere Plugin needs the same Fix Pack as the App. Server.

<sup>4</sup>If you are using 8.5.5 with Fix Pack 9 or 10, set:

````
```

If you are using 8.5.5 with Fix Pack 11, set

````
esl.app.hibernate.transaction.factory_class=org.hibernate.engine.transaction.internal.jta.CMTTransactionFactory.
```

**Red Hat JBoss Enterprise Application Platform**

• 7.1, 7.0, 6.4 — We recommend that you:
  — Install without *Platform Natives*
  — Run with Oracle JDK 1.8.x<sup>6</sup>

<sup>6</sup>The Java Cryptography Extension (JCE) *Unlimited Strength Jurisdiction Policy Files* must be installed for your version of the JDK.

**Wildfly Application Server**

• 17.0 — We recommend that you:
  — Install without *Platform Natives*
  — Run with Oracle JDK 1.8.x<sup>7</sup>

For more on configuring 8.x, see:

https://docs.jboss.org/author/display/WFLY8/General+configuration+concepts

<sup>7</sup>The Java Cryptography Extension (JCE) *Unlimited Strength Jurisdiction Policy Files* must be installed for your version of the JDK.

**Oracle WebLogic Platform**

• 12c (12.1.3) Enterprise or Standard Edition — We recommend that you run with Oracle JDK 1.8.x<sup>8</sup>

<sup>8</sup>The Java Cryptography Extension (JCE) *Unlimited Strength Jurisdiction Policy Files* must be installed for your version of the JDK.
### Table 2-3: Supported (OS, App. Server) & (OS, DB Server) combinations

<table>
<thead>
<tr>
<th>Database Server</th>
<th>OPERATING SYSTEM</th>
<th>Linux</th>
<th>Windows Server</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Oracle Database</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• 19c Enterprise or Standard Edition</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• 12cR2, 12cR1, 11gR2 Enterprise or Standard Edition</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• 11gR2 Enterprise Edition with RAC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• 11gR2 (11.1.0.6.0) Standard Edition with RAC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• 11g Enterprise or Standard Edition</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>IBM DB2 Enterprise Server Edition</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• 11.1, 10.5, 10.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Microsoft SQL Server</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MySQL</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• 5.7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 2-4: Additional supported third-party products

<table>
<thead>
<tr>
<th>ITEM</th>
<th>SUPPORTED ON</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Browser:</strong></td>
<td></td>
</tr>
<tr>
<td><em>Click-to-Sign signing</em></td>
<td></td>
</tr>
<tr>
<td>• Microsoft Internet Explorer 11</td>
<td></td>
</tr>
<tr>
<td>• Mozilla Firefox (latest version)</td>
<td></td>
</tr>
<tr>
<td>• Chrome (latest version)</td>
<td></td>
</tr>
<tr>
<td>• Safari 7 on Mac OS</td>
<td></td>
</tr>
<tr>
<td>ITEM</td>
<td>SUPPORTED ON</td>
</tr>
<tr>
<td>------</td>
<td>--------------</td>
</tr>
</tbody>
</table>
| • Signing with a Certificate | **Browser:**  
| • Signing with an Input Device, where the input device is **not** an iPad or mouse |   
| | **Operating System:**  
| | • Microsoft Windows 8, 7  
| | **Browser:**  
| Signing using the e-Sign iPad or Mouse client | • Microsoft Internet Explorer 11  
| | • Mozilla Firefox (latest version)  
| | • Chrome (latest version)  
| | • Safari on an iPad (second generation or higher) running iOS 5.1.1  
| Signing using the OneSpan Sign Mobile SDK | **Operating System:**  
| | • Apple iOS 7 or higher  

<table>
<thead>
<tr>
<th>ITEM</th>
<th>SUPPORTED ON</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Signing using the Personal Certificate Client</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Browser:</strong></td>
<td></td>
</tr>
<tr>
<td>• Mozilla Firefox (latest version)</td>
<td></td>
</tr>
<tr>
<td>• Chrome (latest version)</td>
<td></td>
</tr>
<tr>
<td>• Microsoft Edge (latest version)</td>
<td></td>
</tr>
<tr>
<td><strong>Operating System:</strong></td>
<td></td>
</tr>
<tr>
<td>• Microsoft Windows 10, 7</td>
<td></td>
</tr>
</tbody>
</table>

This module works with a cryptographic token that provides a certificate and its private key. The token can be a USB stick or a smart card. The latter requires a smart-card reader.

The certificate must be "standard X509 v.3". Also, the flag `nonRepudiation` (aka `contentCommitment`) of the certificate extension `keyUsage` (as defined in RFC 5280) must be ON. In addition, the `signatureAlgorithm` (RFC 5280) should contain an RSA algorithm identifier (we have tested only `SHA1RSA` and `SHA256RSA`).

We have tested this module with the following smart cards: (1) PIV cards; (2) Belgian eID cards; (3) Entrust USB tokens; (4) KPN smart cards (https://www.kpn.com/). These tests were made with the following smart-card readers: (1) VASCO DP 870; (2) VASCO DP 875; (3) VASCO DP 905; (4) Litronic 215; (5) Identiv SCR331; (6) HID Omnikey 3121.

**Note:** Signing with a certificate that lacks `nonRepudiation = ON` is not permitted because such certificates are not intended to be used to sign documents. Signatures produced with such certificates can be disputed in court because they do not imply non-repudiation. One class of certificates with `nonRepudiation = ON` is "Personal CDS/AATL certificates". Many Certificate Authorities (CA) sell such certificates under the name Document Signing Certificates.

**Note:** Although a certificate may conform to our basic requirements, it may have other properties (e.g., Certificate Policies, or an `ExtendedKeyUsage` extension) that prevent it from being used to sign documents. Such properties could cause our software to reject the certificate, or could lead to improper signatures. To minimize this risk, all certificates used with this module must be issued by a CA for the purpose of signing documents.
<table>
<thead>
<tr>
<th>ITEM</th>
<th>SUPPORTED ON</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HTTP Server:</strong></td>
<td></td>
</tr>
<tr>
<td>• Microsoft <em>Internet Information Server</em> (IIS) 10.0, 8.5, 8.0 (Windows OS only)</td>
<td></td>
</tr>
<tr>
<td>• For Windows: Apache HTTP Server 2.2.24. <strong>Warning:</strong> We strongly recommend using the Apache version that is automatically installed with the PDF Document Engine. OneSpan does not test other versions of Apache against the PDF Document Engine.</td>
<td></td>
</tr>
<tr>
<td>• For Linux: Apache HTTP Server 2.4.x (RHEL/CentOS 7), 2.2.x (RHEL/CentOS 6). We recommend using the latest version of Apache (e.g., 2.4.39) that has the most up-to-date security fixes.</td>
<td></td>
</tr>
<tr>
<td><strong>Operating System:</strong></td>
<td></td>
</tr>
<tr>
<td>• Red Hat Enterprise Linux (RHEL) 7 Server (7.0 or higher)</td>
<td></td>
</tr>
<tr>
<td>• Red Hat Enterprise Linux (RHEL) 6 Server (6.3 or higher)</td>
<td></td>
</tr>
<tr>
<td>• CentOS 7.x, 6.x</td>
<td></td>
</tr>
<tr>
<td>• Microsoft Windows Server 2016</td>
<td></td>
</tr>
<tr>
<td>• Microsoft Windows Server 2012 R2</td>
<td></td>
</tr>
<tr>
<td>• Microsoft Windows Server 2012</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> (1) The above HTTP Servers and Operating Systems are 64-bit. (2) If you are using a Load Balancer, we suggest that you disable its <em>retry</em> option. (3) Appendix I describes how to optimize performance if you are using Windows 2012 and IIS 8.</td>
<td></td>
</tr>
<tr>
<td><strong>Gemalto’s Cloud HSM-7 (DPOD)</strong></td>
<td></td>
</tr>
<tr>
<td>• Firmware version 7.1.2 (<em>Data Protection on Demand</em>)</td>
<td></td>
</tr>
<tr>
<td>• The <em>Service Client</em> for this firmware</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> This module will work only if the PDF Document Engine and <em>Service Client</em> use CentOS/RHEL 7.x.</td>
<td></td>
</tr>
<tr>
<td><strong>Amazon’s Cloud HSM</strong></td>
<td></td>
</tr>
<tr>
<td>• AWS CloudHSM Client Software v1.0.14 to v1.1.0</td>
<td></td>
</tr>
<tr>
<td>• The firmware for your software version</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> This module will work only if the PDF Document Engine and CloudHSM Client use CentOS/RHEL 7.x.</td>
<td></td>
</tr>
<tr>
<td><strong>Gemalto’s on-premises HSM-7</strong></td>
<td></td>
</tr>
<tr>
<td>• Firmware version 7.2.0</td>
<td></td>
</tr>
<tr>
<td>• Luna SA Client 7.2.x</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> This module will work if the PDF Document Engine and Luna SA Client use CentOS/RHEL 6.x or 7.x.</td>
<td></td>
</tr>
<tr>
<td>ITEM</td>
<td>SUPPORTED ON</td>
</tr>
<tr>
<td>------</td>
<td>--------------</td>
</tr>
</tbody>
</table>
| Gemalto’s on-premises HSM-6 | - Firmware version 6.10.2 or 6.20.2 (for Luna K-6, 1700 or 7000 series)  
- Luna SA Client 5.2.x to 5.4.x  

**Note:** This module will work if the PDF Document Engine and Luna SA Client use CentOS/RHEL 6.x or 7.x.  

**Note:** If you are using HSM firmware version 6.20.2, you cannot use Luna SA Client 5.2.x. |
| OneSpan Sign Application Frontend’s *Classic User Experience* | **Operating System** (64-bit):  
- Red Hat Enterprise Linux (RHEL) 7.x Server  
- Red Hat Enterprise Linux (RHEL) 6.x Server  
- CentOS 7.x, 6.x  

**HTTP Server:**  
- Apache HTTP Server 2.4.x (RHEL/CentOS 7), 2.2.x (RHEL/CentOS 6)\(^1\)  

\(^1\)RHEL 6 and CentOS 6 can be installed with or without JBoss Core Services 2.4. |
| OneSpan Sign Application Frontend’s *New User Experience* | **Operating System** (64-bit):  
- Red Hat Enterprise Linux (RHEL) 7.x Server  
- Red Hat Enterprise Linux (RHEL) 6.x Server  
- CentOS 7.x, 6.x  

**JavaScript Server:**  
- Node.js, version 10 or higher  

**Recommended Node Process Manager:**  
If you want to enable automatic restarts and process-level Load Balancing, you need:  
- Supervisor ([http://supervisord.org/](http://supervisord.org/)), version 3.1 or higher |
| OneSpan Sign Document Converter | **Operating System:**  
- Microsoft Windows Server 2016  

**Java:**  
- JRE/JDK 8 or higher |
| Connector Framework | **Operating System:**  
- CentOS 7.x, 6.x  
- Red Hat Enterprise Linux (RHEL) 7.x Server  
- Red Hat Enterprise Linux (RHEL) 6.x Server  

**Java:**  
- JRE/JDK 8 or higher (Apache Tomcat 8.x) |
<table>
<thead>
<tr>
<th>ITEM</th>
<th>SUPPORTED ON</th>
</tr>
</thead>
</table>
| Admin Console | **Browser:**  
| | • Microsoft Internet Explorer 11, 10  
| | • Safari 5.x  
| | • Mozilla Firefox (latest version)  
| | • Chrome (latest version)  |
| e-Witness | **Browser:**  
| | • Chrome (latest version)  |
|  | **Java:**  
| | • Java SE Runtime Environment (JRE) 8  
| | • Java SE Development Kit (JDK) 8  |
| Archival Module | **Note:**  
| | • The JRE or JDK must be on the path of the Operating System.  
| | • For the Archival Module: (1) the JRE must be the *Standard Oracle JRE*; (2) the JDK must be the *Standard Oracle JDK*.  
| | • The Java Cryptography Extension (JCE) *Unlimited Strength Jurisdiction Policy Files* must be installed for your version of the JRE or JDK.  
| | • If you are using WebSphere, you can use the IBM JRE that is part of the Application Server.  
| | • If you are using the Archival Module with a DB2 database, you must have disabled progressive streaming on the database server. For more, see the following URLs: (1) [http://www-01.ibm.com/support/docview.wss?uid=swg21983724](http://www-01.ibm.com/support/docview.wss?uid=swg21983724); (2) [http://www-01.ibm.com/support/docview.wss?uid=swg21417865](http://www-01.ibm.com/support/docview.wss?uid=swg21417865)  |
| Installer (see Chapter 3) |  |
| Trusted Service Provider | **Operating System:**  
| | • RHEL/CentOS 7.x  |
User Access Requirements

For all supported Operating Systems and Database Servers (and whether or not you use the Installer):

- The user who installs OneSpan Sign must have READ and WRITE privileges for each Application Server’s Operating System.
- The user who runs the Application Server must have READ and WRITE privileges for each Application Server’s Operating System.
WRITE privileges for the home directories of all OneSpan Sign components (e.g., the OneSpan Sign home directory, the e-Mail Notification Manager™ home directory, etc.).

- All database users must have DELETE, INSERT, SELECT, and UPDATE privileges for all database tables.
- All database users must have SELECT privileges for all database sequences.

**Tenants**

Multitenancy is an architecture in which a single running instance of an application simultaneously serves multiple logically isolated but physically integrated clients. Each client is called a tenant.

OneSpan Sign 5.1 introduced the term tenant as the new name for what previous versions of OneSpan Sign called an Application.

From the perspective of configuration, OneSpan Sign generally works only in tenant mode — i.e., each component must be tenant-aware. The only exception to this rule is the PDF Document Engine. To see how it handles configuration (including the possibility of zero tenants), see Tenant and Non-Tenant Environments (page 112) and Appendix G.

When tenants are supported, each tenant’s configuration is stored remotely. This permits each tenant to be given a different configuration, and that in turn enables multiple configurations per server. For example, with multiple tenants, a single server can offer separate configurations for the mortgage and investment departments of a bank.

With multiple tenants, the same OneSpan Sign deployment can host separate configurations of all the product’s modules. Before version 5.1 of OneSpan Sign, the product did not in fact support full separate configurations per tenant because some settings were global. Now OneSpan Sign supports full separate configurations per tenant.

Each Admin Console action necessarily applies to one of the following:

- The tenant called system (which this guide also calls the System Tenant)
- Another particular e-signature tenant

A System Admin can perform actions that apply at the System Tenant level, while a Tenant Admin can perform actions that apply to their particular tenant. When a tenant is created, it inherits properties from the System Tenant.

For more about creating and managing tenants, see the Platform Administrator’s Guide.
Platform Configuration

Figure 2-2 provides an overview of how the system configures OneSpan Sign Platform components.

The rest of this section discusses the most important elements in Figure 2-2, namely:

- Resource Manager (page 25)
- Admin Console (page 26)
- Client Components (page 26)

Resource Manager

The Master Repository in Figure 2-2 is a File System structure that contains both directories (for components and tenants) and properties files. The properties files specify configuration parameters for various Platform components.

The Resource Manager lies at the heart of the configuration system. It is a Web Service that manages resources (e.g., templates, properties, Data Sources) by handling all read/write requests to the Master Repository.

The Resource Manager uses the JVM system property RESOURCE_PERMANENT_STORAGE to specify the File System path of the Master Repository; see Installing the Resource Manager (page 63).

Just like a database system, the Resource Manager’s Web Service must be up and running before other components can access it.
Admin Console

The Admin Console is a component that enables authorized Admins to configure and maintain the Platform. The user credentials and permissions governing the console are managed by the User Manager. The Admin Console uses the JVM parameter USER_SERVICE_URL to specify the location of the User Manager; see Installing the Admin Console (page 65).

Similarly, the Admin Console uses the JVM parameter RESOURCE_SERVER_LOCATION to specify the location of the Resource Manager.

The Admin Console creates a local cache of the Master Repository, so the console can continue to function for some time even if the Master Repository becomes unavailable. By default, that local cache is the system-dependent temporary-file directory (e.g., /tmp), together with the suffix ac_cache. If you want to change the location of this cache, you must specify the JVM system property RESOURCE_LOCAL_CACHE. Note that the suffix ac_cache will be added to the end of your RESOURCE_LOCAL_CACHE specification.

The directory of this local cache should never be auto-cleaned. Yet on RHEL/CentOS, the system-dependent temporary-file directory is auto-cleaned by default.

Even if you are in a clustered environment, we strongly recommend not having more than one instance of the Admin Console.

The Platform Administrator’s Guide describes how to use the Admin Console.

Client Components

In Figure 2-2, the term Client Component refers to a Platform component (other than the Admin Console or Resource Manager) that must be configured, if only through the default values of certain configuration parameters — e.g., Core Component, Scheduler, Notifier, e-Witness, PDF Document Engine.

Figure 2-2 shows generic Client Components 1 and 2, whose configuration is managed by the other elements in the figure.

Client Components use the JVM parameter RESOURCE_SERVER_LOCATION to specify the location of the Resource Manager. If that parameter specifies a File System instead of a regular network URL, then: (1) the entire Resource Manager system will be bypassed; (2) the Client Component must become its own configuration system. This should not be done in a production environment or in a clustered environment. Indeed, it should be done only in a One Box Staging/Demo environment.

Like the Admin Console, each Client Component creates a local cache of the Master Repository. By default, that local cache is the system-dependent temporary-file directory (e.g., /tmp), together with the suffix esep. If you want to
change the location of this cache, you must specify the JVM system property
`RESOURCE_LOCAL_CACHE`.

⚠️ The directory of this local cache should never be auto-cleaned. Yet on
RHEL/CentOS, the system-dependent temporary-file directory is auto-cleaned by
default.

In a vertical cluster, `RESOURCE_LOCAL_CACHE` must be configured separately
for each node (i.e., each node must point to its own local cache). This ensures that
no conflicts will occur among multiple nodes. The same rule applies to multiple
stand-alone installations of the Platform on a single machine.

After creating a cache, a Client Component waits a default period of 900 seconds,
and then tries to communicate with the Master Repository. If you want to change
this default period, you must change the value of the JVM parameter
`CACHE_LIFETIME_SECONDS`.

# Deployment Topology

Figure 2-3 illustrates the recommended deployment topology for the REST
Integration Model. In tests, this topology was found to deliver the highest
throughput. Note that it’s recommended to deploy the Platform and Application
on the same host/server, but on different JVMs.

Figure 2-3 reflects the fact that, whether you’re installing the Platform or the
Application, we strongly recommend a **horizontally scaled environment**. That is,
multiple instances of a given component should be installed on different physical
machines and/or JVMs (Java virtual machines), where: (1) each machine can
independently receive and respond to requests; (2) the machines operate in
parallel.

A horizontally scaled environment:

- Increases throughput and thus increases efficiency
- Offers protection against hardware failures
- Provides load balancing and process failover

From the perspective of deployment, our system uses both **homogeneous** clusters
(with similar instances) and **heterogeneous** clusters. Both of them support horizontal
scalability.

As Figure 2-3 illustrates:

- The Platform was designed to be deployed on a **homogeneous** cluster.
- The Application is currently designed to be deployed on a **heterogeneous**
  cluster (for example, there should be only one instance of the *OneSpan
  Sign Scheduler*). Many Application Servers do not support
  heterogeneous clusters. For those servers, something like such a cluster
can be manually created as a set of load-balanced individual instances.
Deployment Road Map

A typical deployment of OneSpan Sign can generally be thought of as proceeding via the following grand steps:

1. **Step 1**: Review Figure 2-4, which illustrates the recommended Deployment Road Map.

2. **Step 2**: Prepare your Database System. Then create all relevant databases (see Appendix A).

   If you are using WebSphere, Figure 2-5 and the procedure immediately following it indicate how to accomplish Step 3 below.

   If you are using JBoss/Wildfly, Figure 2-6 and the procedure immediately following it indicate how to accomplish Step 3 below.
3. **Step 3**: Prepare an Application Server environment; see Preparing an App Server Environment (page 312). Then do the following:
   • If you want to use the Installer, use it to install components (see Chapter 3).
   • If you do not want to use the Installer:
     a. Manually install all required Platform components except the PDF Document Engine (see Chapters 4-10).
     b. Manually install any desired optional Platform components (see Chapters 12-14).
     c. Manually install all required Application components and any desired optional Application components except the OneSpan Sign Application Frontend and OneSpan Sign Document Converter. For details, see Chapter 15, Figure 2-4, Figure 2-5, and Figure 2-6.

4. **Step 4**: Prepare machines for your PDF Document Engines, and then install them (see Chapter 11).

5. **Step 5**: Prepare a machine for the OneSpan Sign Application Frontend, and then install it. For details, see Deploying the Transaction Status API (page 195).

6. **Step 6**: Prepare a machine for the OneSpan Sign Document Converter, and then install it. For details, see Installing the OneSpan Sign Document Converter (page 228).

7. **Step 7**: Install and configure any Application hardware components you want (see Chapter 16).

8. **Step 8**: Configure the system (see Chapter 17 and Chapter 18).

In this section’s figures, two numbers separated by an ellipsis (x..y) represent the sequence of integers between the two numbers. The number of instances of a given component in a single deployment can be any number in its associated x..y sequence. For example:

- **1..n** indicates that: (1) there must be at least one instance of the associated component; (2) there is no upper limit to its number of instances.
- **1..1** indicates that there should be only one instance of the associated component, even if you are in a clustered environment. Such components cannot be load-balanced.

In Figure 2-4, the OneSpan Sign Scheduler, OneSpan Sign BackOffice, Resource Manager, and Admin Console are 1..1. This implies that each has only a single point of failure. This is acceptable because until the failed instance is restored, the system will continue to function (e.g., messages in queues are persisted; BackOffice persists data in the database; each Platform instance runs with cached data of the Resource Manager; the Admin Console is used only to change configurations in Resource Manager data).

Note that any component in this section’s figures associated with an expression of
the form $0..n$ or $1..n$ can be horizontally scaled in a cluster of n machines.

$1..1$ under the Admin Console in Figure 2-5 suggests that it cannot have multiple instances. This is not strictly true. However, we’ve written $1..1$ in this case because a single instance of the Admin Console is strongly recommended.

The $1..n$ under the Core Database in Figure 2-5 is intended to convey that you can use multiple database schemas in a tenant environment (see Creating the Core Database in Appendix A).
To implement the deployment strategy for WebSphere, do the following using Figure 2-5 as a guide:

1. Create 1..n Application Servers (e.g., Server1 and Server2). These will be used for the Application components on Servers 1 and 2 of Figure 2-5.

   Instead of the previous step’s default deployment strategy for the Application, you could create a cluster of 1..n members (e.g., ApplicationCluster consisting of Member1 and Member2) for the Application components. This would be useful especially if you are using WebSphere SIB default messaging, since it would facilitate adding messaging engines on the fly to increase scalability.

   Warning: Such a cluster should not include certain Application components (OneSpan Sign BackOffice, OneSpan Sign Scheduler), since a single instance of them outside the cluster will suffice.

2. Create a cluster of 1..n members (e.g., PlatformCluster consisting of Member1 and Member2). This will be used for the Platform components.

3. Create a single Application Server (a stand-alone Application Server) on one node. This will be used for the Admin Console and the Resource Manager.
4. If you are using WebSphere SIB default messaging:
   a. Create two separate buses -- one for the Application, one for the Platform.
   b. Assign the servers from Step 1 and the PlatformCluster from Step 2 as bus members to appropriate buses.
   c. Create and assign:
      • The Application's destinations to the Server1 bus member
      • The Platform's destinations to the PlatformCluster bus member
5. Deploy the Application's components (see Figure 2-5).
6. Create JMS and JDBC resources for the Application at the Cell scope level.
7. Deploy the Platform's components at the Cluster scope level.
8. Create JMS and JDBC resources for the Platform at the Cluster scope level.

   This procedure ends here.

---

**Deployment Strategy for JBoss EAP 6**

To implement the deployment strategy for JBoss or Wildfly, do the following using Figure 2-6 as a guide:

1. Create five JBoss or Wildfly Standalone Application Servers (e.g.,
2. Deploy the Application’s components for each server.
3. Create JMS and JDBC resources for the Application on each relevant server.
4. Deploy the Platform’s components.
5. Create JMS and JDBC resources for the Platform on each relevant server.

This procedure ends here.

Packaging

The packaging described in this section includes third-party software. All licenses for the third-party products are up-to-date.

A OneSpan Sign release is packaged according to the structure described in this section. The top directory contains the Release Notes, as well as three main sub-directories:

1. **Client** — Contains OneSpan Sign client-side software that’s required when an input device is used for signing.
2. **Documentation** — Contains: (1) all relevant OneSpan Sign guides in PDF format; (2) third-party licenses.
5. **Server** — Contains OneSpan Sign server-side software.

The **Server** directory contains the following sub-directories:

- **Application Backend Components** — The installation files for the components of the Application Backend.
- **Application Frontend Components** — The installation files for the components of the Application Frontend.
- **Core Components** — Includes multiple sub-directories and *.ear files:
- **Installer** — Contains software for the OneSpan Sign Installer application.
- **Optional Modules** — Contains files for the following optional modules: Archival Module, OneSpan Sign Mobile SDK, e-Witness.
PART II: THE INSTALLER

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>3: Using the Installer</td>
<td>35</td>
</tr>
</tbody>
</table>
CHAPTER 3: Using the Installer

This chapter describes automated ways of installing and uninstalling the following OneSpan Sign components on WebSphere, JBoss, or Wildfly by using an Installer:

Required Platform Components:

1. Core Component
2. e-Mail Notification Manager
3. Audit Service
4. Resource Manager
5. User Manager
6. Event Manager
7. Admin Console (required to configure the system)

Optional Platform Components:

8. e-Witness
9. Archival Module

Required Application Components:

10. OneSpan Sign Application Backend
11. OneSpan Sign Authentication
12. OneSpan Sign Scheduler
13. OneSpan Sign BackOffice
14. BackOffice Support
15. Transaction Status API (required for the Application and e-Witness)

Optional Application Components:

16. OneSpan Sign e-Notary Validation
17. OneSpan Sign Equifax
18. OneSpan Sign Single Sign-On
19. OneSpan Sign Mobile SDK
20. OneSpan Sign URL Mapper
21. OneSpan Sign Document Converter

Later chapters describe how to install and uninstall the above components manually.

If you plan to use the OneSpan Sign Application Frontend and/or OneSpan Sign Document Converter and/or any Application Connector, after using the Installer you must manually install those components (see Chapter 15).
Installing Components

The Installer cannot deploy: (a) one required component (PDF Document Engine); (b) certain optional components (e.g., Archival Module, Trusted Service Provider).

The rest of the chapter describes:

- Installing Components (page 36)
- Uninstalling Components (page 51)

Installing Components

If you are using JBoss/Wildfly and deploying both OneSpan Sign and the e-Vault Manager, you must install OneSpan Sign and the e-Vault Manager on two different JBoss/Wildfly servers.

Even if you are in a clustered environment, there should be only one instance of the OneSpan Sign Scheduler and only one instance of OneSpan Sign BackOffice.

The recommended practice is to install the Resource Manager on its own Application Server. Even if you are in a clustered environment, there should be only one instance of this component.

This section describes an automated way of installing various components by using an Installer. Later chapters describe how to install the same components manually.

If you are upgrading to OneSpan Sign 7.2 from an earlier version of the product, you may wish to contact OneSpan.

Figure 2-5 illustrates the strategy employed by the Installer for WebSphere. Figure 2-6 illustrates the strategy employed by the Installer for JBoss or Wildfly.

Prerequisites

- Your environment meets all relevant requirements in Chapter 2.
- You have performed the relevant procedure from Preparing an App Server Environment (page 312).
- If you plan to install the Core Component, its General Prerequisites (page 71) have been satisfied.
- If you plan to install the Application components, the procedure called Creating the Application Database (page 288) has been performed.
- If you plan to install the e-Mail Notification Manager™, the procedure called Creating the e-Mail Notification Manager Database (page 283) has been performed.
- If you plan to install the Audit Service, the procedure called Creating the Audit Database (page 287) has been performed.
- If you plan to install the User Manager, all prerequisites from Installing...
the User Manager (page 58) have been satisfied.

• If you plan to install the Archival Module, all prerequisites from Installing the Archival Module (page 174) have been satisfied.

• All Application Server settings have been committed.

⚠️ If not all Application Server settings have been committed, the Installer will be unresponsive, and you will have to manually force it to close and restart.

• If you are using WebSphere in a clustered environment, you have mapped or mounted each node under the machine where the Deployment Manager is running.

**Action**

Step 2 below asks you to specify various passwords that appear in Table 3-1. The Installer automatically encrypts these passwords.

**To install OneSpan Sign components using the Installer:**

1. On the machine where the Application Server is installed, go to the directory that contains the "Installer" sub-directory (you can find a copy of that sub-directory in the software provided by OneSpan).

2. Create an environmental configuration file "EnvConfig.xml" in the directory "Installer/EnvConfig" (the top directory here is the "Installer" directory in the Packaging section of Chapter 2).

We recommend that you create your configuration file by customizing one of the sample files in the directory "EnvConfig/samplesConfigFiles". **For WebSphere,** the **template sample files** "EnvConfig for Multi-Install WAS SIB.xml" and "EnvConfig for Multi-Install WAS MQ.xml" follow the default recommended topology and strategy described in Chapter 2. **For JBoss or Wildfly,** start with the **template sample file** "EnvConfig for JBoss6.xml" or "EnvConfig for WildFly.xml".

For a description of the XML tags in the configuration file "EnvConfig.xml", see Table 3-1. You must use these tags to customize the installation to your software environment. Tags or attributes you don't need can safely be left blank or removed from the file.

⚠️ The values specified in Table 3-1 are case-sensitive.

⚠️ The "EnvConfig.xml" file contains some parameters that specify directory locations. If you do not use the UNC (Universal Naming Convention) format for paths, be sure to specify each path in a manner that is local to the machine of the component that will use the path.
The default values in Table 3-1 reflect values assigned in the sample files in the directory "EnvConfig/samplesConfigFiles". The Installer uses those default values. If you want to change those defaults, you must edit those values in your final "EnvConfig.xml".

3. Copy the license file "silanisLicense.xml" into the directory "Installer/EnvConfig".

4. Run the following command (making appropriate substitutions for the placeholders):


   [1] = A comma-separated list within a single pair of double quotes (" ") of the features to be installed. To install: (1) Core Component, type Core; (2) e-Mail Notification Manager™, type CoreScheduler, CoreNotifier; (3) Resource Manager, type Resource Manager; (4) User Manager, type UserManager; (5) Event Manager, type Event Manager; (6) Admin Console, type Admin Console; (7) Archival Module, type Database Archival Module; (8) e-Witness, type eWitness; (9) e-SL UX Backend, type Backend; (10) e-SL BackOffice, type BackendBO; (11) e-SL Scheduler, type BackendBatch; (12) e-SL e-Notary Validation, type BackendEno; (13) e-SL Equifax, type BackendEquifax; (14) e-SL Authentication, type BackendAuth; (15) e-SL Single Sign-On, type BackendSSO; (16) Transaction Status API, type Transaction Status; (17) e-SL URL Mapper, type BackendUrlMapper; (18) OneSpan Sign Document Converter, type BackendOOConverter; (19) all features, type *.

   If you type Core in [1], the Audit Service will be installed automatically.

   [2] = The exact version number of OneSpan Sign to be installed (e.g., 6.5.2). This argument is required. The Installer can be used to install any version of OneSpan Sign from 4.6 to 7.2. **WARNING:** If you are using WebSphere 9, the specified version number must be 7.2.9.

   If you are using WebSphere, you may encounter a JVM exception (java/lang/OutOfMemoryError) while the above command is running. If so, you must increase the heap size for the Java machine used by the wsadmin. To learn how to do so, please consult relevant IBM documentation.

   If you are using JBoss EAP or Wildfly, the Installer will create its own dedicated XML configuration file named esep.xml. In doing so, the Installer will use as a template the default XML configuration file named standalone-full.xml.

   By default, the Installer tests the credentials of the database needed to configure the Data Source. This operation can be disabled using the optional switch -notestdb on the above command line.
5. If you chose to install the Core Component, e-Witness, or Admin Console, configure HTTPS access to them.

6. If you are using WebSphere in a clustered environment, restart the Node Agents.

! We strongly recommend that you perform Steps 4-7 of the Deployment Road Map in Chapter 2 before you perform the remaining steps below.

7. If you are using a Microsoft SQL Server database whose version number is 2012 or higher, add the following system variable:

   `esl.app.hibernate.dialect=com.silanis.esl.hibernate.dialect.CUSTOMSqlServerDialect`

8. Restart all Application Servers, starting with the Administration JVM (see Figure 2-4).

   After restarting, many errors will appear in your log files. These should be harmless, appearing only because you have not yet configured the system.

9. Configure the Platform by performing the procedure in Chapter 17.

10. Configure the Application by performing the procedure in Chapter 18.

   If you don’t configure Table 18-1’s parameters in Step 10, log-file errors will ask you to do so when you try to restart the Application Servers in Step 11.

11. Clear your log files. Then restart all Application Servers, starting with the Administration JVM (see Figure 2-4).

12. If you deployed the Core Component, verify its installation by going to the following URL:

   • `https://<server>:<port>/esep/rest/application.wadl`

   Here `<server>` denotes the server where the Core Component is installed, and `<port>` denotes the port it uses. You should be able to see this file’s contents, or be able to download the file.

13. If you deployed e-Witness, verify its installation by using a browser to go to the following URL:

   • `https://<server>:<port>/evidence`

   Here `<server>` denotes the server where e-Witness is installed, and `<port>` denotes the port it uses. You should see the logon page of e-Witness.

14. If you deployed the Admin Console, verify its installation by viewing the login page of the Web-based Admin Console at the following URL:

   • `https://<server>:<port>/esep/admin-console/main/index.htm`
15. If you deployed the User Manager, verify its installation by navigating to the following URL:
   • https://<server>:<port>/user/rest/v1/application.wadl

   Type the default credentials (username = system.admin and password = password). A WADL page should appear.

16. If you deployed the Transaction Status API, verify its installation by navigating to the following URL:
   • https://<server>:<port>/tsapi/services/soap?wsdl

17. Optional: If you are using JBoss or Wildfly, you deployed the Core Component or e-Mail Notification Manager™, and you want to change its logging behavior, consult Appendix B. If you deployed e-Witness, verify its installation by using a browser to go to the following URL:

18. The Installer has not been updated in some time. Until it is, you must perform the following steps:

   a. If you are using WebSphere, for each JMS queue: (1) from the WebSphere Administration Console, go to Resources > JMS > Queues > Queue name > Advanced properties; (2) select the checkbox Append RFH version 2 headers to messages sent to this destination Message.

   b. If you are using JBoss or Wildfly:
      • Edit the Application file /standalone/configuration/esep.xml by adding the part of the following code that twice specifies max-post-size:

        ```
        <subsystem xmlns="urn:jboss:domain:undertow:1.2">
          <buffer-cache name="default"/>
          <server name="default-server">
            <http-listener name="default" max-post-size="26214400" socket-binding="http"/>
            <https-listener name="default-ssl" max-post-size="26214400" socket-binding="https" security-realm="SslRealm"/>
          </server>
        </subsystem>
        ```

      • Edit the Platform file .../standalone/configuration/esep.xml by adding the part
of the following code that specifies `max-post-size`:

```xml
<subsystem xmlns="urn:jboss:domain:undertow:1.2">
  <buffer-cache name="default"/>
  <server name="default-server">
    <http-listener name="default" max-post-size="31457280" socket-binding="http"/>
  </server>
</subsystem>
```

This procedure ends here.

You must configure the installed components. See Chapters 21 and 22 of this guide, as well as the *Platform Administrator’s Guide* and *Application Administrator’s Guide*.

If you plan to use the *OneSpan Sign Application Frontend* and/or *OneSpan Sign Document Converter* and/or any connector, you must manually install those components using Chapter 15).

This procedure ends here.
Table 3-1: Parameters in the configuration file "EnvConfig.xml"

<table>
<thead>
<tr>
<th>PARAMETER (XML TAG)</th>
<th>DESCRIPTION</th>
<th>VALUES OR NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;environment_configuration&gt;</td>
<td>Root tag for the &quot;EnvConfig.xml&quot; file</td>
<td>Possible values:</td>
</tr>
</tbody>
</table>
| encryption | Parameter that determines if all passwords are currently encrypted (before the Installer is run) | • true  
• false (default)  
Note: The Installer always sets this parameter to true after it reads the configuration file. Thus the Installer always encrypts all passwords. If you set all passwords to be readable (i.e., not encrypted), set this parameter to false. However, note that the Installer will subsequently encrypt all passwords and reset this parameter to true. |
### INSTALLING COMPONENTS

**Note:** You can use the Installer to install/uninstall multiple modules in multiple destinations by doing either of the following: (1) if the modules have the same software environment, define them by using the comma-separated list in the next row; (2) if the modules have different software environments, define them by using the *module* tag multiple times.

<table>
<thead>
<tr>
<th>PARAMETER (XML TAG)</th>
<th>DESCRIPTION</th>
<th>VALUES OR NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;module&gt;</td>
<td>Start tag for the definition of one or more modules</td>
<td>Possible values for the modules in the comma-separated list are the <em>NAME</em> values in the first column of Table 3-2. The last column of Table 3-2 specifies which tags in <em>this</em> table must be defined for the module in question.</td>
</tr>
<tr>
<td>name</td>
<td>A comma-separated list of the modules to be installed/uninstalled</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PARAMETER (XML TAG)</th>
<th>DESCRIPTION</th>
<th>VALUES OR NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;os&gt;</td>
<td>Start tag for Operating System parameters</td>
<td>Possible values:</td>
</tr>
<tr>
<td>name</td>
<td>Name of the Operating System used by the Core Component</td>
<td>• windows (default)</td>
</tr>
<tr>
<td>install_home</td>
<td>OneSpan Sign home directory</td>
<td>• linux</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default: c:/Program Files/Silanis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: This parameter is used to create JVM arguments such as AWS_HOME, REVIEWER_HOME, and USER_HOME.</td>
</tr>
</tbody>
</table>
### Installing Components

**<installdir>** Start tag for installation directory parameters

<table>
<thead>
<tr>
<th>PARAMETER (XML TAG)</th>
<th>DESCRIPTION</th>
<th>VALUES OR NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>dir</td>
<td>OneSpan Sign target installation directory for each node mapping (including local installations)</td>
<td>Default: c:/Program Files/Silanis</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This is the physical representation where the product will be installed.</td>
<td></td>
</tr>
</tbody>
</table>

**</installdir>** End tag for installation directory parameters

**</os>** End tag for Operating System parameters

**<database>** Start tag for parameters that specify Data Sources for OneSpan Sign

<table>
<thead>
<tr>
<th>PARAMETER (XML TAG)</th>
<th>DESCRIPTION</th>
<th>VALUES OR NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the currently defined Data Source</td>
<td>Possible values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Core</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DocumentRetriever</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• eWitness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Scheduler</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CoreArchive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• TSAPI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Audit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• UserMgr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Backend</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Backoffice</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ReportingPool</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• UrlMapper</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Possible values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• db2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• oracle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• sqlserver</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• mysql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more, see Table 3-2 and Table 3-3.</td>
</tr>
<tr>
<td>type</td>
<td>Type of database server for the current Data Source</td>
<td>Possible values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• db2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• oracle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• sqlserver</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• mysql</td>
</tr>
</tbody>
</table>

**username** Username used to connect to the database server for the current Data Source

**password** Password used to connect to the database server for the current Data Source

**hostname** Hostname or IP address of the "computer" on which the database server for the current Data Source is running
<table>
<thead>
<tr>
<th>PARAMETER (XML TAG)</th>
<th>DESCRIPTION</th>
<th>VALUES OR NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>port</td>
<td>Port number of the database server for the current Data Source</td>
<td></td>
</tr>
<tr>
<td>schema</td>
<td>The name of the schema under which tables for the current Data Source were created</td>
<td>No default</td>
</tr>
<tr>
<td>dbname</td>
<td>Name of the database for the current Data Source</td>
<td>Note: If you specify this parameter, you must leave service unspecified.</td>
</tr>
<tr>
<td>service</td>
<td>Name of the Oracle™ RAC used for the current Data Source</td>
<td>Note: If you specify this parameter, you must leave dbname unspecified.</td>
</tr>
<tr>
<td>jaasAlias</td>
<td>Name of the JAAS (Java™ Authentication and Authorization Service) alias used (and created by the Installer) for the current Data Source [WebSphere only]</td>
<td>Default: esep</td>
</tr>
<tr>
<td>version</td>
<td>Version of the Oracle Database [Oracle and WebSphere only]</td>
<td>Possible values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 10g</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 11g</td>
</tr>
<tr>
<td>schemaVersion</td>
<td>Version of the OneSpan Sign database schema that will be queried by the Transaction Status API</td>
<td>Default: 5.0</td>
</tr>
</tbody>
</table>

</database>  
End tag for the OneSpan Sign database parameters

<userm>  
Start tag for the User Manager’s API parameters

<table>
<thead>
<tr>
<th>PARAMETER (XML TAG)</th>
<th>DESCRIPTION</th>
<th>VALUES OR NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>URL of the User Manager’s API Web service</td>
<td>This parameter must be in the following format:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="https://host:port/user/rest/v1">https://host:port/user/rest/v1</a></td>
</tr>
</tbody>
</table>

</userm>  
End tag for the User Manager’s API parameters
<table>
<thead>
<tr>
<th>PARAMETER (XML TAG)</th>
<th>DESCRIPTION</th>
<th>VALUES OR NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;resM&gt;</code></td>
<td>Start tag for the Resource Manager’s location</td>
<td>No default</td>
</tr>
<tr>
<td><code>&lt;appserver&gt;</code></td>
<td>Start tag for Application Server parameters</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>Name of the Application Server</td>
<td>Possible values:</td>
</tr>
<tr>
<td></td>
<td>Note: The appropriate tag must be defined (e.g., if the value of this parameter is <code>jboss6</code>, the <code>jboss6</code> tag must be defined).</td>
<td>• websphere • jboss6 • wildfly</td>
</tr>
<tr>
<td>install_dir</td>
<td>Directory that contains the Application Server</td>
<td>Possible values:</td>
</tr>
<tr>
<td></td>
<td>• AppServer (WebSphere) • The top-level directory (JBoss or Wildfly)</td>
<td></td>
</tr>
<tr>
<td><code>&lt;jboss6&gt;</code></td>
<td>Start tag for JBoss 6 EAP parameters</td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>Version of JBoss you are using</td>
<td>Possible values:</td>
</tr>
<tr>
<td></td>
<td>• 6</td>
<td></td>
</tr>
<tr>
<td>server</td>
<td>Server directory for the JBoss Application Server you are using</td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>standalone</td>
<td></td>
</tr>
<tr>
<td>&lt;/jboss6&gt;</td>
<td>End tag for JBoss 6 EAP parameters</td>
<td></td>
</tr>
<tr>
<td><code>&lt;wildfly&gt;</code></td>
<td>Start tag for Wildfly parameters</td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>Version of Wildfly you are using</td>
<td>Possible values:</td>
</tr>
<tr>
<td></td>
<td>• 8</td>
<td></td>
</tr>
<tr>
<td>server</td>
<td>Server directory for the Wildfly Application Server you are using</td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>standalone</td>
<td></td>
</tr>
<tr>
<td>&lt;/wildfly&gt;</td>
<td>End tag for Wildfly parameters</td>
<td></td>
</tr>
</tbody>
</table>
### PARAMETER (XML TAG) | DESCRIPTION | VALUES OR NOTES
--- | --- | ---
<websphere> | Start tag for WebSphere parameters | 

**messaging** | Type of messaging (= queue type) | Possible values:
  - sib
  - mq

**installedEar** | The directory WebSphere will use to deploy components | Valid directory

<cell> | Start tag for target-cell parameters | 

**name** | Name of the target cell where OneSpan Sign will be installed | 

**busName** | Name of the Service Integration Bus OneSpan Sign will use to send JMS messages | Default: awsBus

<loadbalancer> | Start tag for IBM HTTP Server (IHS) parameters (part of the current cell). Note: The XML definition must exactly match your Load Balancer topology. | 

**name** | Name of the IBM HTTP Server (IHS) that will load-balance OneSpan Sign | Default: webserver1

**node** | Name of the node where the IBM HTTP Server (IHS) is running | Default: xxxNode01

</loadbalancer> | End tag for IBM HTTP Server (IHS) parameters | 

<dm> | Start tag for Deployment Manager parameters (part of the current cell) | 

**name** | Name of the Deployment Manager Profile or Stand-Alone Application Server where OneSpan Sign will be installed | Default: Dmgr01

**username** | Admin username used to log on to the Application Server | Default: admin

**password** | Password for the Admin username | 

**path** | Directory that contains the Deployment Manager Profile or Stand-Alone Profile | Default: c:/Program Files/IBM/WebSphere/AppServer/profiles/Dmgr01

</dm> | End tag for Deployment Manager parameters | 

<mqserver> | Start tag for WebSphere MQ Server parameters (part of the current cell) | 

**mgrName** | Queue Manager Name of the WebSphere MQ Server | Default: AWS_QM

**host** | Host name or IP address of the "computer" on which the WebSphere MQ Server is running | Default: localhost
<table>
<thead>
<tr>
<th>PARAMETER (XML TAG)</th>
<th>DESCRIPTION</th>
<th>VALUES OR NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>port</td>
<td>Port number of the WebSphere MQ Server</td>
<td>Default: 1414</td>
</tr>
<tr>
<td>channel</td>
<td>Channel used by the WebSphere MQ Server</td>
<td>Default: SYSTEM.ADMIN.SVRCONN</td>
</tr>
<tr>
<td>jaasAlias</td>
<td>Name of the JAAS (Java™ Authentication and Authorization Service) alias [WebSphere only]</td>
<td><strong>Note</strong>: The Installer will create and use the jaasAlias.</td>
</tr>
<tr>
<td>username</td>
<td>Username used to connect to the MQ server</td>
<td></td>
</tr>
<tr>
<td>password</td>
<td>Password used to connect to the MQ server</td>
<td></td>
</tr>
</tbody>
</table>

```xml
</mqserver>  
```

**End tag for WebSphere MQ Server parameters**

```xml
<server>  
```

**Start tag for WebSphere Application Server parameters (part of the current cell)**** Note**: You can use the server tag or the cluster tag, but not both.

<table>
<thead>
<tr>
<th>name</th>
<th>Name of the Websphere Application Server where OneSpan Sign will be deployed</th>
<th>Default: server1</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>Name of the Node where the Application Server is running</td>
<td>Default: xxxNode01</td>
</tr>
</tbody>
</table>

```xml
</server>  
```

**End tag for WebSphere Application Server parameters**

```xml
<cluster>  
```

**Start tag for cluster parameters (part of the current cell)**** Note**: You can use the server tag or the cluster tag, but not both.

The XML definition must exactly match your cluster topology.

<table>
<thead>
<tr>
<th>name</th>
<th>Name of the cluster created inside the WebSphere Application Server DM profile</th>
<th>Default: awsCluster</th>
</tr>
</thead>
</table>
| policy             | Current policy regarding the messaging engines for the cluster where OneSpan Sign is running | Possible values:  
  - HA
  - SCALABILITY
  - SCALABILITY_HA |

```xml
<member>  
```

**Start tag for cluster-member parameters**

<table>
<thead>
<tr>
<th>name</th>
<th>Name of the Websphere Application Server where OneSpan Sign will be deployed</th>
<th></th>
</tr>
</thead>
</table>
Table 3-2: Modules that can be installed/uninstalled

<table>
<thead>
<tr>
<th>INSTALLER TARGET NAME</th>
<th>MODULE NAME</th>
<th>REQUIRED TAGS FROM TABLE 3-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>Core Component (including the Audit Service)</td>
<td>os, database (name = Core, Audit, DocumentRetriever), appserver</td>
</tr>
<tr>
<td>CoreNotifier*</td>
<td>Notifier</td>
<td>os, appserver</td>
</tr>
<tr>
<td>CoreScheduler*</td>
<td>Scheduler</td>
<td>os, database (name = Scheduler), appserver</td>
</tr>
<tr>
<td>Resource Manager*</td>
<td>Resource Manager</td>
<td>os, appserver</td>
</tr>
<tr>
<td>User Manager*</td>
<td>User Manager</td>
<td>os, database (name = UserMgr), appserver</td>
</tr>
<tr>
<td>Event Manager*</td>
<td>Event Manager</td>
<td>os, appserver</td>
</tr>
<tr>
<td>Admin Console</td>
<td>Admin Console (the Installer installs both versions)</td>
<td>os, appserver, userm, database (name = Core)</td>
</tr>
<tr>
<td>eWitness</td>
<td>e-Witness</td>
<td>os, database (name = Core, tsapi, eWitness), appserver</td>
</tr>
<tr>
<td>Database Archival Module</td>
<td>Archival Module</td>
<td>os, database (name = Core; also name = CoreArchive), appserver</td>
</tr>
<tr>
<td>Backend</td>
<td>OneSpan Sign UX Backend</td>
<td>os, database (name = Backend, ReportingPool), appserver</td>
</tr>
<tr>
<td>BackendBO</td>
<td>OneSpan Sign BackOffice</td>
<td>os, database (name = Backoffice, ReportingPool), appserver</td>
</tr>
<tr>
<td>BackendInternal</td>
<td>BackOffice Support</td>
<td>os, appserver</td>
</tr>
<tr>
<td>BackendBatch</td>
<td>OneSpan Sign Scheduler</td>
<td>os, appserver</td>
</tr>
</tbody>
</table>
Table 3-3: Mapping between Installer names & databases

<table>
<thead>
<tr>
<th>INSTALLER DATABASE TARGET NAME</th>
<th>DATABASE IDENTITY</th>
<th>REFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>Core Database</td>
<td>Appendix A</td>
</tr>
<tr>
<td>Document Retriever</td>
<td>Information required to retrieve document content from a database</td>
<td>None</td>
</tr>
<tr>
<td>e-Witness</td>
<td>Core Database or any Archive Database</td>
<td>Appendix A</td>
</tr>
<tr>
<td>Scheduler</td>
<td>e-Mail Notification Manager Database</td>
<td>Appendix A</td>
</tr>
<tr>
<td>CoreArchive</td>
<td>Archive Database</td>
<td>Appendix A</td>
</tr>
<tr>
<td>TSAPI</td>
<td>Core Database</td>
<td>Appendix A</td>
</tr>
<tr>
<td>Admin Console</td>
<td>Core Database</td>
<td>Appendix A</td>
</tr>
<tr>
<td>Audit</td>
<td>Audit Database</td>
<td>Appendix A</td>
</tr>
<tr>
<td>UserMgr</td>
<td>User Manager Database</td>
<td>Appendix A</td>
</tr>
<tr>
<td>Backend</td>
<td>OneSpan Sign Application Database</td>
<td>Appendix A</td>
</tr>
<tr>
<td>Backoffice</td>
<td>OneSpan Sign Application Database</td>
<td>Appendix A</td>
</tr>
</tbody>
</table>

*Module requires the Core Component*
To uninstall OneSpan Sign components using the Installer:

1. On the machine where the Application Server is installed, go to the directory that contains the "Installer" sub-directory.
2. Edit the environmental configuration file "EnvConfig.xml" in the directory "Installer/EnvConfig".

For a description of the XML tags in the configuration file "EnvConfig.xml", see Table 3-1. You must use these tags to customize the "uninstall" to your software environment. Tags or attributes you don't need can safely be left blank or removed from the file.

The values specified in Table 3-1 are case-sensitive.

The "EnvConfig.xml" file contains some parameters that specify directory locations. If you do not use the UNC (Universal Naming Convention) format for paths, be sure to specify each path in a manner that is local to the machine of the component that will use the path.

The default values in Table 3-1 reflect values assigned in the sample files in the directory "EnvConfig/samplesConfigFiles". The Installer uses those default values. If you want to change those defaults, you must edit those values in your final "EnvConfig.xml".

If during installation you changed the name of the EAR/WAR file (or the name of the Application name used by WebSphere) for a given component, ensure that those changes are reflected in the next step.
3. Run the following command (making appropriate substitutions for the placeholders):

```java
```

[1] = A comma-separated list within a single pair of double quotes (" ") of the features to be uninstalled. To uninstall: (1) Core Component, type Core; (2) e-Mail Notification Manager™, type CoreScheduler, CoreNotifier; (3) Resource Manager, type Resource Manager; (4) User Manager, type User Manager; (5) Event Manager, type Event Manager; (6) Admin Console, type Admin Console; (7) Archival Module, type Database Archival Module; (8) e-Witness, type eWitness; (9) e-SL UX Backend, type Backend; (10) e-SL BackOffice, type BackendBO; (11) e-SL Scheduler, type BackendBatch; (12) e-SL e-Notary Validation, type BackendEno; (13) e-SL Equifax, type BackendEquifax; (14) e-SL Authentication, type BackendAuth; (15) e-SL Single Sign-On, type BackendSSO; (16) Transaction Status API, type Transaction Status; (17) e-SL URL Mapper, type BackendUrlMapper; (18) OneSpan Sign Document Converter, type BackendOOConverter; (19) all features, type *.

If you type Core in [1], the Audit Service will be uninstalled automatically.

[2] = The exact version number of OneSpan Sign to be uninstalled (e.g., 6.1.8). This argument is required. The Installer can be used to uninstall any version of OneSpan Sign that was installed by the Installer. **WARNING**: If you are using WebSphere 9, the specified version number must be 7.2.9.

This procedure ends here.
# PART III: REQUIRED PLATFORM SOFTWARE

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4: Deploying the Audit Service</td>
<td>54</td>
</tr>
<tr>
<td>5: Deploying the User Manager</td>
<td>58</td>
</tr>
<tr>
<td>6: Deploying the Resource Manager</td>
<td>63</td>
</tr>
<tr>
<td>7: Deploying the Admin Console</td>
<td>65</td>
</tr>
<tr>
<td>8: Deploying the Core Component</td>
<td>71</td>
</tr>
<tr>
<td>9: Deploying the Event Manager</td>
<td>89</td>
</tr>
<tr>
<td>10: Deploying the e-Mail Notification Manager</td>
<td>94</td>
</tr>
<tr>
<td>11: Deploying PDF Document Engines</td>
<td>112</td>
</tr>
</tbody>
</table>
CHAPTER 4: Deploying the Audit Service

The Audit Service is an OneSpan Sign module that can create reports about the past activities of OneSpan Sign users and Admin Console users. This feature was introduced in OneSpan Sign 5.1.

This chapter describes:

- Installing the Audit Service (page 54)
- Uninstalling the Audit Service (page 56)

The Audit Service is required by the User Manager, Resource Manager, and e-Witness.

Installing the Audit Service

This section describes how to manually install the Audit Service. Chapter 3 describes an alternate, automated way of doing so if you are using WebSphere, JBoss, or Wildfly.

Prerequisites

- Your environment meets all relevant requirements in Chapter 2.
- You have performed the relevant procedure from Preparing an App Server Environment (page 312).
- The procedure Creating the Audit Database (page 287) has been performed.

Action

To install the Audit Service:

- Create a Data Source for the Audit Service using: (i) the parameters in Table 4-1; (ii) the appropriate procedure from Creating a JDBC Provider (page 316).

The user account for the Data Source should have read/write access to the Audit Database.

This procedure ends here.

Table 4-1: Audit Service Data Source parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Audit Service Data Source</td>
<td>Any character string</td>
</tr>
<tr>
<td>WebSphere calls this parameter the Data source name</td>
<td></td>
</tr>
</tbody>
</table>
### Installing the Audit Service

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>JNDI Name associated with the Audit Service Data Source</td>
<td>This name must be: <code>audit/jdbc/ConnectionPool</code></td>
</tr>
<tr>
<td>Transaction Isolation Level</td>
<td>READ_COMMITTED</td>
</tr>
<tr>
<td></td>
<td>For example, for WebSphere add a datasource custom property called <code>webSphereDefaultIsolationLevel</code>, and set its value to 2.</td>
</tr>
<tr>
<td>Current Schema</td>
<td>The name of the schema under which tables for the Audit Database were created in this procedure’s prerequisites (using Appendix A)</td>
</tr>
<tr>
<td></td>
<td>This parameter is needed only if those tables were not under the same schema as the user defined in the current Data Source, and if you are using WebSphere. If these conditions apply, do the following:</td>
</tr>
<tr>
<td></td>
<td>• If you are using DB2, set the value of the datasource custom property called <code>currentSchema</code> to the value of this parameter.</td>
</tr>
<tr>
<td></td>
<td>• If you are using a DB2 or Oracle Database, ensure that the file <code>audit-settings.properties</code> (accessible via the Admin Console) includes the following property specification:</td>
</tr>
<tr>
<td></td>
<td>```</td>
</tr>
<tr>
<td></td>
<td>audit.schema=&lt;schema&gt;</td>
</tr>
<tr>
<td></td>
<td>```</td>
</tr>
<tr>
<td></td>
<td>where <code>&lt;schema&gt;</code> is the value of this parameter (see Table B-5 of the Platform Administrator’s Guide).</td>
</tr>
<tr>
<td>Use Java Context</td>
<td>The value of <code>&lt;use-java-context&gt;</code> must be <code>false</code>.</td>
</tr>
</tbody>
</table>
| | **Note:** This parameter is needed only if you are using JBoss or Wildfly.
To uninstall the Audit Service:

1. Delete the Data Source for the Audit Database.
2. Delete the JDBC Provider for the Audit Database (provided it’s not being used by any other application).
3. If you are using WebSphere, JBoss, or Wildfly, and if you copied any JDBC driver files for the Audit Database when the Audit Service was installed, delete them (provided they’re not being used by any other application).
4. If you are using WebLogic, if you were not using an Oracle database when the Audit Service was installed, and if you copied any JDBC driver files for the Audit Database at that time, delete them (provided they’re not being used by any other application).
5. Perform the following related steps:
   a. Make a backup copy of the Audit Database.

If you do not have a backup copy of this database, you will permanently lose all its data.
b. Delete the Audit Database and all its contents.

   This procedure ends here.
CHAPTER 5: Deploying the User Manager

The User Manager is an OneSpan Sign module that manages user credentials and permissions. This module was introduced in OneSpan Sign 5.1.

This chapter describes:

• Installing the User Manager (page 58)
• Uninstalling the User Manager (page 61)

The User Manager is required by the Admin Console.

Installing the User Manager

This section describes how to manually install the User Manager. Chapter 3 describes an alternate, automated way of doing so if you are using WebSphere, JBoss, or Wildfly.

Prerequisites

• Your environment meets all relevant requirements in Chapter 2.
• The following procedures have been performed:
  • Creating the User Manager Database (page 287)
  • Installing the Audit Service (page 54)

Action

To install the User Manager:

1. On the Application Server host, create a directory that will serve as the User Manager home directory.
2. Configure the Application Server with the following JVM system property:
   • USER_HOME=[user-manager-home-directory]
3. Copy the file "user-hibernate.properties" from the software provided by OneSpan Sign to the directory [user-manager-home-directory].
4. Edit the file "user-hibernate.properties" by replacing all contents inside curly brackets {} — as well as those brackets themselves — with contents applicable to your Application Server and database (see the examples included in this file).
5. If you are installing the User Manager in a clustered Application Server environment, do one of the following:
   • Copy the User Manager home directory created in the preceding steps to each node in the cluster.
   • Ensure that the User Manager home directory is on a network
drive that can be accessed by all nodes in the cluster.

6. Create a Data Source for the User Manager Database, using: (i) the parameters in Table 5-1; (ii) the appropriate procedure from Creating a JDBC Provider (page 316). Ensure that Container Managed Persistence (CMP) is not selected.

![Note]
The user account for the Data Source should have read/write access to the User Manager Database.

7. Inside the Application Server, deploy the file "eSL-UserManager.ear" provided by OneSpan.

8. Restart the Application Server.

9. Verify the installation by doing the following:
   a. Start the Application Server software, and verify that during startup no errors were logged in the User Manager log or the Application Server log.
   b. Use the Application Server software to verify that the User Manager was properly deployed.
   c. Navigate to the following URL:
      
      - `https://<server>:<port>/user/rest/v1/application.wadl`

      Type the default credentials (username = `system.admin` and password = `password`). A WADL page should appear.

This procedure ends here.

**Table 5-1: User Manager's Data Source settings**

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the User Manager’s Data Source</td>
<td>Any valid character string</td>
</tr>
<tr>
<td>JNDI Name associated with the User Manager’s Data Source</td>
<td>This name must be &quot;user/jdbc/ConnectionPool&quot;.</td>
</tr>
<tr>
<td>Transaction Isolation Level</td>
<td>For example, for WebSphere add a datasource custom property called <code>webSphereDefaultIsolationLevel</code>, and set its value to 2.</td>
</tr>
</tbody>
</table>
### Current Schema

The name of the schema under which tables for the User Manager Database were created in this procedure’s prerequisites (using Appendix A).

This parameter is needed only if those tables were not under the same schema as the user defined in the current Data Source, and if you are using WebSphere. If these conditions apply, do the following:

- If you are using a DB2 or Oracle Database, ensure that the file `user-hibernate.properties` (see Step 4) includes the following property specification:
  ```
  hibernate.default_schema={schema}
  ```
  where `{schema}` is the value of this parameter.

### Driver Class

**Note**: This parameter is needed only if you are using JBoss or Wildfly.

The value of `<driver-class>` must be as follows for the three possible databases:

- **Oracle** — `oracle.jdbc.driver.OracleDriver`
- **DB2** — `com.ibm.db2.jcc.DB2Driver`
- **SQLServer** — `com.microsoft.sqlserver.jdbc.SQLServerDriver`
- **MySQL** — `com.mysql.jdbc.Driver`

### Definition of the JBoss/Wildfly driver referenced by the User Manager Data Source

Possible values:

- `oracle.ojdbc` (Oracle)
- `ibm.db2` (DB2)
- `com.microsoft` (MS SQL Server)
- `mysql` (MySQL)
Uninstalling the User Manager

This section describes how to manually uninstall the User Manager. Chapter 3 describes an alternate, automated way of doing so if you are using WebSphere, JBoss, or Wildfly.

To uninstall the User Manager:

1. Inside the Application Server, uninstall the "eSL User Manager" application.

   If you are using JBoss or Wildfly, this step means deleting the file "eSL-UserManager.ear".

2. Delete the activation specification created when the User Manager was installed.

3. Delete the Data Source for the User Manager Database.

4. Delete the JDBC Provider for the User Manager Database (provided it’s not being used by any other application).

5. If you are using WebSphere, JBoss, or Wildfly, and if you copied any JDBC driver files for the User Manager Database when the User Manager was installed, delete them (provided they’re not being used by any other application).

6. If you are using WebLogic, if you were not using an Oracle database when the User Manager was installed, and if you copied any JDBC driver files for the User Manager Database at that time, delete them (provided they’re not being used by any other application).

7. Perform the following related steps (if you are using a cluster, perform these steps on each Application Server host):
   a. Make a backup copy of the home directory of the User Manager.

      If you do not have a backup copy of this directory, you will permanently lose all its data.

   b. Delete the User Manager home directory and all its contents.

8. Delete the contents of the User Manager Database (but not the database itself).

9. If the User Manager Database is not part of the Core Database, perform the following related steps:
   a. Make a backup copy of the User Manager Database.

      If you do not have a backup copy of this database, you will permanently lose all its data. However, don’t delete this database if it is part of the Core Database.

   b. Delete the User Manager Database and all its contents.
10. Restart the Application Server.

   This procedure ends here.
CHAPTER 6: Deploying the Resource Manager

The Resource Manager is an OneSpan Sign module that manages resources (e.g., templates, properties, Data Sources).

This chapter describes:

- Installing the Resource Manager (page 63)
- Uninstalling the Resource Manager (page 64)

The Resource Manager is required by the Admin Console.

Installing the Resource Manager

This section describes how to manually install the Resource Manager. Chapter 3 describes an alternate, automated way of doing so if you are using WebSphere, JBoss, or Wildfly.

Even if you are in a clustered environment, there should be only one instance of the Resource Manager.

We recommend that you start the Resource Manager before all other components that are installed on the same server as the Resource Manager.

Prerequisites

- Your environment meets all relevant requirements in Chapter 2.
- The Application Server is installed and running on the Application Server machine.

The recommended practice is to install the Resource Manager on its own Application Server.

- You have performed the relevant procedure from Preparing an App Server Environment (page 312).
- The procedure Installing the Audit Service (page 54) has been performed.

Action

To install the Resource Manager:

1. On the Application Server host, create the Resource Manager’s home directory that will serve as its storage location.
2. Inside the Application Server, deploy the "eSL-ResourcesManager.ear" file provided by OneSpan. This will install the Resource Manager component.

Even if you are in a clustered environment, there should be only one instance of this component.

3. Configure the Application Server with the following Java Virtual Machine (JVM) system property:

```
RESOURCE_PERMANENT_STORAGE = [resource-manager-home-directory]
```

It’s a "best practice" to regularly back up this directory.

4. Restart the Application Server.

This procedure ends here.

**Uninstalling the Resource Manager**

This section describes how to manually uninstall the Resource Manager. Chapter 3 describes an alternate, automated way of doing so if you are using WebSphere, JBoss, or Wildfly.

**To uninstall the Resource Manager:**

1. Inside the Application Server, uninstall the application "eSL-Resources Manager".

   If you are using JBoss or Wildfly, this step means deleting the file "eSL-ResourceManager.ear" from the deployment directory.

2. Unset the generic JVM arguments that were set when the Resource Manager was installed.

3. Perform the following related steps:
   a. Make a backup copy of the Resource Manager’s home directory.

   If you do not have a backup copy of this directory, you will permanently lose all its data.

   b. Delete the Resource Manager’s home directory, and all its contents.

4. Restart the Application Server.

   This procedure ends here.
CHAPTER 7: Deploying the Admin Console

The 5.1 version of OneSpan Sign introduced a new Admin Console that completely replaced the Admin Console that was part of previous versions of OneSpan Sign.

This section describes:

- Installing the Admin Console (page 65)
- Uninstalling the Admin Console (page 69)

The Admin Console is required by the Scheduler, Notifier, PDF Document Engine, Event Manager, and Transaction Status API (the last item only if it’s configured to use the Resource Manager — see the Transaction Status API Deployment Guide).

Installing the Admin Console

This section describes how to manually install the Admin Console. Chapter 3 describes an alternate, automated way of doing so if you are using WebSphere, JBoss, or Wildfly.

Even if you are in a clustered environment, we strongly recommend not having more than one instance of the Admin Console.

The new Admin Console has two alternative User Interfaces that have identical functionality:

- A text-based interface (TUI)
- A Web-based interface

Their installation steps differ, but both interfaces can be installed in the same environment. At least one of them must be installed to enable the proper configuration of OneSpan Sign.

Prerequisites

- A Java runtime environment is installed. We recommend that you use the official Oracle JDK. The Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files must be installed for your version of the JDK.
- The following procedures have been sequentially performed:
  a. Installing the User Manager (page 58)
  b. Installing the Resource Manager (page 63)

The TUI Admin Console can be run on any machine with: (1) connectivity to the User Manager and Resource Manager; (2) an installed Java runtime. The Web-based Admin Console runs on the Application Server.
Action

To install the Admin Console:

1. If you want to install the TUI Admin Console:
   a. Copy the Admin Console Text directory provided by OneSpan to your local machine.
   b. Open the file "AdminConsole.properties", which can be found in your installation directory.
   c. Specify the following parameters by making them reference the Resource Manager’s server and User Manager’s server, respectively:

   ```
   resource.server.location = https://<Resource_Master_Base_HOST>:<port>/resources-master/storage
   user.management.location = https://<User_Manager_Base_HOST>:<port>/user/rest/v1
   ```
   d. Save and close the file "AdminConsole.properties".

2. If you want to install the Web-based Admin Console:
   a. On the machine that will run the Admin Console, ensure that the following JVM parameters are set:
      • `AWS_HOME = <eSL_HOME>`
      • `RESOURCE_SERVER_LOCATION = https://<Resource_Master_Base_URL>:<port>/resources-master/storage`
      • `USER_SERVICE_URL = https://<User_Master_Server>:<port>/user/rest/v1`
   b. If you want to change the local cache location (which by default is the system-dependent temporary-file directory, together with the suffix `ac_cache`), configure the Application Server with the following Java Virtual Machine (JVM) system property:
      ```
      RESOURCE_LOCAL_CACHE = <path to local cache>
      ```
      ! The directory of this local cache should never be auto-cleaned. Yet on RHEL/CentOS, the system-dependent temporary-file directory is auto-cleaned by default.
   c. Create a Data Source for the Core Database using: (i) the parameters in Table 7-1; (ii) the appropriate procedure from
Creating a JDBC Provider (page 316). Ensure that Container Managed Persistence (CMP) is **not** selected.

The user account for the Data Source should have read/write access to the Core Database.

d. If you want to change the default expiration time of the local cache (which is 900 seconds), configure the Application Server with the following Java Virtual Machine (JVM) system property:

   - `CACHE_LIFETIME_SECONDS = <N>`

e. Inside the Application Server, deploy the file "eSL-AdminConsole.ear" provided by OneSpan.

f. Verify that the installation was successful by viewing the login page of the Web-based Admin Console at the following URL:

   ```
   ```

3. If you are using WebLogic, the Application Server may perform its own default **HTTP Basic Authentication**. This depends on your WebLogic version (e.g., it’s true of version 12.1.3.0.0). If this is true, you must disable that authentication, since HyperBolic implements its own **HTTP Basic Authentication**. To disable the unwanted authentication:

   a. In the directory `Oracle_Home/ser_projects/domains/base_domain/<your_domain>/config/`, locate and open the file `config.xml`.

   b. Between the tags `<security-configuration>` and `</security-configuration>`, insert the following line:

   ```
   <enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>
   ```

   c. Save and close `config.xml`.

   This procedure ends here.

### Table 7-1: OneSpan Sign Data Source parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name of the OneSpan Sign Data Source</strong></td>
<td>WebSphere calls this parameter the <strong>Data source name</strong>. Any character string</td>
</tr>
<tr>
<td><strong>JNDI Name associated with the OneSpan Sign Data Source</strong></td>
<td>This name must be: <code>awsng/jdbc/ConnectionPool</code></td>
</tr>
</tbody>
</table>
### Transaction Isolation Level

- **Description:** For example, for WebSphere add a datasource custom property called `webSphereDefaultIsolationLevel`, and set its value to 2.

- **Values:**

<table>
<thead>
<tr>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>The name of the schema under which tables for the Core Component were created in this procedure’s prerequisites (using Appendix A)</td>
<td></td>
</tr>
<tr>
<td>This parameter is needed only if those tables were not under the same schema as the user defined in the current Data Source, and if you are using WebSphere. If these conditions apply, do the following:</td>
<td></td>
</tr>
<tr>
<td>• If you are using DB2, set the value of the datasource custom property called <code>currentSchema</code> to the value of this parameter.</td>
<td></td>
</tr>
<tr>
<td>• If you are using a DB2 or Oracle Database, ensure that the file <code>hibernate.properties</code> (accessible via the Admin Console) includes the following property specification:</td>
<td></td>
</tr>
<tr>
<td>hibernate.default_schema=&lt;schema&gt;</td>
<td></td>
</tr>
<tr>
<td>where <code>&lt;schema&gt;</code> is the value of this parameter (see Table C-17 of the Platform Administrator’s Guide).</td>
<td></td>
</tr>
</tbody>
</table>

### Use Java Context

- **Description:** The value of `<use-java-context>` must be `false`.

- **Note:** This parameter is needed only if you are using JBoss or Wildfly.
Uninstalling the Admin Console

To uninstall the Admin Console:

1. If you want to uninstall the TUI Admin Console, delete the directory that contains the Admin Console application.
2. If you want to uninstall the Web-based Admin Console:
   a. Unset the generic JVM arguments that were set when the Admin Console was installed.
   b. Inside the Application Server, uninstall the application "eSL Admin Console".

   If you are using JBoss or Wildfly, this step means deleting the file "eSL-AdminConsole.ear" from the deployment directory.

   This procedure ends here.
CHAPTER 8: Deploying the Core Component

The Core Component is the primary component that supplies the core functionality of OneSpan Sign. It runs on the Application Server.

This chapter describes:

- Installing the Core Component (page 71)
- Uninstalling the Core Component (page 84)

Installing the Core Component

If you are upgrading to OneSpan Sign 7.x from an earlier version of OneSpan Sign, you may wish to contact OneSpan.

When you install the Core Component, the Audit Service is installed automatically.

To install the Core Component:

1. Ensure that all General Prerequisites (page 71) are satisfied.
2. Perform one of the following procedures (depending on your Application Server):
   - Installing the Core Component on WebSphere (page 72)
   - Installing the Core Component on JBoss or Wildfly (page 79)
   - Installing the Core Component on WebLogic (page 81)

General Prerequisites

Whatever Application Server you use, your installation of the Core Component should begin by ensuring that the following prerequisites are met.

- Your environment meets all relevant requirements in Chapter 2.
- The Application Server is installed and running on the Application Server machine.
- The procedure called Creating the Core Database in Appendix A has been performed.
- The Core Component must be installed on the same Application Server instances as the Event Manager.
- If you are using Linux, the System Administrator has added the following line (note its initial asterisk) to the file "limits.conf", which is in the directory "/etc/security":

  * hard nofile 4096

This increases the number of file handles available to a OneSpan Sign
user for any executable process they might run.

**Installing the Core Component on WebSphere**

This section describes how to manually install the Core Component on WebSphere. Chapter 3 describes an alternate, automated way of doing so.

**Prerequisites**

- All General Prerequisites (page 71) are satisfied.

**Action**

To install the Core Component on WebSphere:

1. On the Application Server host, create a directory that will serve as the OneSpan Sign home directory.
2. Into this `[eSL_HOME]` directory, copy the file `silanisLicense.xml` provided by OneSpan.
3. Configure the Application Server with the following Java Virtual Machine (JVM) system properties:
   - `AWS_HOME = [eSL_HOME]`
   - `RESOURCE_SERVER_LOCATION = https://<Resource Manager Server>:<port>/resources-master/storage`
   - `javax.xml.transform.TransformerFactory = org.apache.xalan.processor.TransformerFactoryImpl`
4. If you want to change the default expiration time of the local cache (which is 900 seconds), configure the Application Server with the following Java Virtual Machine (JVM) system property:
   - `CACHE_LIFETIME_SECONDS = <N>`
   - **Step 5 and Step 6 are required if you are using vertical clustering, which is not recommended.**
5. If you want to change the local cache location (which by default is the system-dependent temporary-file directory, together with the suffix `esep`), configure the Application Server with the following Java Virtual Machine (JVM) system property:
   - `RESOURCE_LOCAL_CACHE = <path to local cache>`
   - The directory of this local cache should never be auto-cleaned. Yet on RHEL/CentOS, the system-dependent temporary-file directory is auto-cleaned by default.
6. The Core Component submits frequent requests to the PDF Document Engine. Each request has an ID that looks something like this: `fc70e50d-977b-48d4-8e60-b3474da10028-2984-<string>`. 
Installing the Core Component on WebSphere

Here <string> is a character string whose default value is the host name of the machine where the installation is taking place. If multiple Core Component instances are installed on that machine, using that default value may be confusing because no ID can uniquely identify its associated instance. If you want to avoid that confusion, configure the Application Server with the following JVM system property:

- `aws.jvm.instance.name = <Core-Component-instance name>`

For example, if you set that property to "awsinstancenode1", a typical ID will look something like this: bb3e30-3638-4b69-a8e3-b17e9608c2c2-awsinstancenode1.

7. In the `[eSL_HOME]` directory, create one directory called "logs" and another directory called "logging".

8. Configure the Apache log4j logging facility by doing the following:
   a. Into the "logging" directory, copy the file "log4j.properties" from the "Server/Core Components/Resource" directory of the software provided by OneSpan.
   b. Edit the "log4j.properties" file you just copied, configuring it appropriately for your needs.
      
      For detailed information about the Apache log4j logging facility, go to:
      
      - [http://logging.apache.org/log4j/1.2/](http://logging.apache.org/log4j/1.2/)
   c. Configure the Application Server with the following JVM system property:
      
      - `log4j.configuration = file:[eSL_HOME]/logging/log4j.properties`

9. If you are installing the Core Component in a clustered Application Server environment, do one of the following:
   
   - Copy the OneSpan Sign home directory created in the preceding steps to each node in the cluster.
   - Ensure that the OneSpan Sign home directory is on a network drive that can be accessed by all nodes in the cluster.

10. Create a Data Source for the Core Database using: (i) the parameters in Table 7-1; (ii) the appropriate procedure from Creating a JDBC Provider (page 316). Ensure that Container Managed Persistence (CMP) is not selected.
    
    The user account for the Data Source should have read/write access to the Core Database.

11. If any of your e-signature processes must fetch documents from a database (instead of from the File System), create a Data Source for the Document Retriever Pool Database using: (i) the parameters in Table 8-5; (ii) the appropriate procedure from Creating a JDBC.
Installing the Core Component on WebSphere

Provider (page 316). Ensure that Container Managed Persistence (CMP) is not selected.

The user account for the Data Source should have READ access to the Document Retriever Pool Database.

12. If you are using WebSphere SIB default messaging:
   a. Create a bus for OneSpan Sign, and assign as its “members” the servers or clusters on which the Core Component will run. If you wish, configure bus security.

   Steps 13a-13f use the bus created in Step 12a.

   b. Create the following destinations within the bus:
      • ErrorQueueDestination
      • PostOnlineQueueDestination
      • ConcatenateDocumentsQueueDestination
      • NotifierResponseQueueDestination

13. To complete the configuration of JMS resources:
   a. Create a Queue Connection Factory with JNDI name "jms/QueueConnectionFactory".
   b. Create the following persistent JMS queues:
      • ErrorQueue (use the parameters in Table 8-1)
      • PostOnlineQueue (use the parameters in Table 8-2)
      • ConcatenateDocumentsQueue (use the parameters in Table 8-3)
      • NotifierResponseQueue (use the parameters in Table 8-4)

      For each of the above queues: (1) from the WebSphere Administration Console, go to Resources > JMS > Queues > Queue name > Advanced properties; (2) select the checkbox Append RFH version 2 headers to messages sent to this destination Message.

   c. Create the following activation specification for the ErrorQueue:
      • Name: ActivationSpecError
      • JNDI name: jms/ActivationSpecError
      • Destination JNDI name: jms/queue/ErrorQueue

   d. Create the following activation specification for the PostOnlineQueue:
      • Name: ActivationSpecPostOnline
      • JNDI name: jms/ActivationSpecPostOnline
      • Destination JNDI name: jms/queue/PostOnlineQueue
e. Create the following activation specification for the ConcatenateDocumentsQueue:
   • Name: ActivationSpecMergeDoc
   • JNDI name: jms/ActivationSpecMergeDoc
   • Destination JNDI name: jms/queue/ConcatenateDocumentsQueue

f. Create the following activation specification for the NotifierResponseQueue:
   • Name: ActivationSpecNotificationResponse
   • JNDI name: jms/ActivationSpecNotificationResponse
   • Destination JNDI name: jms/queue/NotifierResponseQueue

14. Inside the Application Server, deploy the "eSL-Core.ear" file provided by OneSpan.

15. If you are using "out-of-process events" (i.e., general system events that are independent of any particular e-signature process), configure the following Java system parameter (default = false):
   • -Dinclude.outofprocess.event = true

   This parameter is a flag that determines if out-of-process events will be included in e-signature processes.

16. If you have a remote connection to the JMS queue of the Scheduler and/or if you want to change the Queue Connection Factory name from that specified in Step 13 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "schedulerJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator’s Guide).

17. If you have a remote connection to the JMS queue of the Notifier and/or if you want to change the Queue Connection Factory name from that specified in Step 13 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "notifierJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator’s Guide).

18. Configure HTTPS access to the Core Component.

19. Perform the procedure Enabling Centralized Apache log4j Logging (page 291) in Appendix B.

20. Restart the Application Server.

21. Verify the above installation by going to the following URL:
   • https://<server>:<port>/esep/rest/application.wadl
Here `<server>` denotes the server where the Core Component is installed, and `<port>` denotes the port it uses. You should be able to see this file’s contents, or be able to download the file.

To enable the Core Component to function properly, you must also: (a) install the Event Manager, Scheduler, and Notifier (see Chapters 9-10); (b) configure the Core Component using the Platform Administrator’s Guide (see Tables C-1 and C-7 of that guide).

This procedure ends here.

### Table 8-1: Error Queue parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Error Queue</td>
<td>ErrorQueue</td>
</tr>
<tr>
<td>WebSphere calls this parameter the Base queue name.</td>
<td>APPROVEIT.ERROR.QUEUE</td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
<td>APPROVEIT.ERROR.QUEUE</td>
</tr>
<tr>
<td>JNDI name associated with the Error Queue</td>
<td>The JNDI name must contain the queue name as follows. If the queue name is &quot;ErrorQueue&quot;, the JNDI name must be &quot;jms/queue/ErrorQueue&quot;.</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
<td>This value must be Persistent.</td>
</tr>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
<td>2</td>
</tr>
<tr>
<td>Queue to which an error is delivered when the maximum number of retries is reached</td>
<td>none</td>
</tr>
</tbody>
</table>

### Table 8-2: Post-Online Queue parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Post-Online Queue</td>
<td>PostOnlineQueue</td>
</tr>
<tr>
<td>WebSphere calls this parameter the Base queue name.</td>
<td>APPROVEIT.POSTONLINE.QUEUE</td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
<td>APPROVEIT.POSTONLINE.QUEUE</td>
</tr>
<tr>
<td>JNDI name associated with the Post-Online Queue</td>
<td>The JNDI name must contain the queue name as follows. If the queue name is &quot;PostOnlineQueue&quot;, the JNDI name must be &quot;jms/queue/PostOnlineQueue&quot;.</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
<td>Persistent</td>
</tr>
</tbody>
</table>
### Table 8-3: Concatenate Documents Queue parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
<td>2</td>
</tr>
<tr>
<td>WebSphere MQ and WebSphere call this parameter the <strong>Backout threshold</strong>.</td>
<td></td>
</tr>
<tr>
<td>Queue to which an error is delivered when the maximum number of retries is reached</td>
<td>The error queue</td>
</tr>
<tr>
<td>WebSphere MQ and WebSphere call this parameter the <strong>Backout requeue queue</strong>.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Concatenate Documents Queue</td>
<td>ConcatenateDocumentsQueue</td>
</tr>
<tr>
<td>WebSphere calls this parameter the <strong>Base queue name</strong>.</td>
<td></td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
<td>APPROVEIT.MERGEDOC.QUEUE</td>
</tr>
<tr>
<td>JNDI name associated with the Concatenate Documents Queue</td>
<td>The JNDI name must contain the queue name as follows. If the queue name is &quot;ConcatenateDocumentsQueue&quot;, the JNDI name must be &quot;jms/queue/ConcatenateDocumentsQueue&quot;.</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
<td>Persistent</td>
</tr>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
<td>1</td>
</tr>
<tr>
<td>WebSphere MQ and WebSphere call this parameter the <strong>Backout threshold</strong>.</td>
<td></td>
</tr>
</tbody>
</table>

### Table 8-4: Notifier Response Queue parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Notifier Response Queue</td>
<td>NotifierResponseQueue</td>
</tr>
<tr>
<td>WebSphere calls this parameter the <strong>Base queue name</strong>.</td>
<td></td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
<td>APPROVEIT.NOTIFIER.RESPONSE.QUEUE</td>
</tr>
<tr>
<td>JNDI name associated with the Notifier Response Queue</td>
<td>The JNDI name must contain the queue name as follows. If the queue name is &quot;NotifierResponseQueue&quot;, the JNDI name must be &quot;jms/queue/NotifierResponseQueue&quot;.</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
<td>Persistent</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>VALUES</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
<td>1</td>
</tr>
<tr>
<td>WebSphere MQ and WebSphere call this parameter the <em>Backout threshold</em>.</td>
<td></td>
</tr>
</tbody>
</table>

**Table 8-5: Data Source parameters for the Document Retriever Pool DB**

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Data Source</td>
<td>Any character string</td>
</tr>
<tr>
<td>WebSphere calls this parameter the <em>Data source name</em>.</td>
<td></td>
</tr>
<tr>
<td>JNDI Name associated with the Data Source</td>
<td>This name must be: <code>awsng/jdbc/DocumentRetrieverPool</code></td>
</tr>
<tr>
<td>Transaction Isolation Level</td>
<td><code>READ_COMMITTED</code></td>
</tr>
<tr>
<td>For example, for WebSphere add a datasource custom property called</td>
<td><code>webSphereDefaultIsolationLevel</code>, and set its value to 2.</td>
</tr>
<tr>
<td>Use Java Context</td>
<td>The value of <code>&lt;use-java-context&gt;</code> must be <code>false</code>.</td>
</tr>
<tr>
<td>Note: This parameter is needed only if you are using JBoss or Wildfly.</td>
<td></td>
</tr>
<tr>
<td>The Driver Class</td>
<td></td>
</tr>
<tr>
<td>Note: This parameter is needed only if you are using JBoss or Wildfly.</td>
<td></td>
</tr>
<tr>
<td>The value of <code>&lt;driver-class&gt;</code> must be as follows for the three possible</td>
<td></td>
</tr>
<tr>
<td>databases:</td>
<td></td>
</tr>
<tr>
<td>Driver Class</td>
<td></td>
</tr>
<tr>
<td><code>Oracle</code> — <code>oracle.jdbc.driver.OracleDriver</code></td>
<td></td>
</tr>
<tr>
<td><code>DB2</code> — <code>com.ibm.db2.jcc.DB2Driver</code></td>
<td></td>
</tr>
<tr>
<td><code>SQLServer</code> — <code>com.microsoft.sqlserver.jdbc.SQLServerDriver</code></td>
<td></td>
</tr>
<tr>
<td><code>MySQL</code> — <code>com.mysql.jdbc.Driver</code></td>
<td></td>
</tr>
<tr>
<td>Non-transactional data source</td>
<td>If you are using WebSphere 8.5.x, this check box must be <em>Enabled</em>.</td>
</tr>
</tbody>
</table>
Installing the Core Component on JBoss or Wildfly

This section describes how to manually install the Core Component on JBoss or Wildfly. Chapter 3 describes an alternate, automated way of doing so.

Prerequisites

- All General Prerequisites (page 71) are satisfied.

Action

To install the Core Component on JBoss or Wildfly:

1. On the Application Server host, create a directory that will serve as the OneSpan Sign home directory.
2. Into this [eSL_HOME] directory, copy the file silanisLicense.xml provided by OneSpan.
3. In the [eSL_HOME] directory, create one directory called "logs" and another directory called "logging".
4. Configure the Application Server with the following JVM system property:
   - AWS_HOME = [eSL_HOME]
   - RESOURCE_SERVER_LOCATION = https://<Resource Manager Server>:<port>/resources-master/storage
5. If you want to change the default expiration time of the local cache (which is 900 seconds), configure the Application Server with the following Java Virtual Machine (JVM) system property:
   - CACHE_LIFETIME_SECONDS = <N>
6. If you want to change the local cache location (which by default is the system-dependent temporary-file directory, together with the suffix esep), configure the Application Server with the following Java Virtual Machine (JVM) system property:
   - RESOURCE_LOCAL_CACHE = <path to local cache>

The directory of this local cache should never be auto-cleaned. Yet on RHEL/CentOS, the system-dependent temporary-file directory is auto-cleaned by default.
7. Even if none of your e-signature processes needs to fetch documents from a database (instead of from the File System), you must create a Data Source for the Document Retriever Pool Database using: (i) the parameters in Table 8-5; (ii) the appropriate procedure from Creating a JDBC Provider (page 316).

The user account for the Data Source should have READ access to the Document Retriever Pool Database. Make the Data Source point to the Core Database (even if your e-signature processes don’t need to fetch documents from a database).

8. Create JMS resources for the Core Component by doing the following:
   a. Add a new Queue Connection Factory with JNDI name "jms/QueueConnectionFactory".
   b. Create the following persistent JMS queues:
      • ErrorQueue (use the parameters in Table 8-1)
      • PostOnlineQueue (use the parameters in Table 8-2)
      • ConcatenateDocumentsQueue (use the parameters in Table 8-3)
      • NotifierResponseQueue (use the parameters in Table 8-4)

9. Inside the Application Server, deploy the file "eSL-Core.ear".

10. If you want to enable SOAP security for the PDF Document Engine:
    a. In the [eSL_HOME] directory, create the sub-directory /axis2repo/modules.
    b. Into that modules directory, copy the files "rampart-1.6.2.mar" and "rahas-1.6.2.mar" from the "Server/Core Components/Thirdparty/docengine_soap_client" directory of the software provided by OneSpan.
    c. Open the startup script for JBoss or Wildfly, and ensure that the following Java option is present:
        -Daxis2.repo="[eSL_HOME]/axis2repo"
    d. If you changed the startup script, save and close it.

For more on SOAP security, see Chapter 11’s section called Enabling Soap Security.

11. If you are using "out-of-process events" (i.e., general system events that are independent of any particular e-signature process), configure the following Java system parameter (default = false):
    • -Dinclude.outofprocess.event = true

This parameter is a flag that determines if out-of-process events will be included in e-signature processes.

12. If you have a remote connection to the JMS queue of the Scheduler and/or if you want to change the Queue Connection Factory name from that specified in Step 8 and/or if you want to use credentials for the
queues and the Queue Connection Factory, configure "schedulerJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator's Guide).

13. If you have a remote connection to the JMS queue of the Notifier and/or if you want to change the Queue Connection Factory name from that specified in Step 8 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "notifierJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator's Guide).

14. Configure HTTPS access to the Core Component.

15. Perform the procedure Enabling Centralized Apache log4j Logging (page 291) in Appendix B.

16. Start the Application Server.

17. Verify the above installation by going to the following URL:
   - https://<server>:<port>/esep/rest/application.wadl

   Here <server> denotes the server where the Core Component is installed, and <port> denotes the port it uses. You should be able to see this file's contents, or be able to download the file.

To enable the Core Component to function properly, you must also:
(a) install the Event Manager, Scheduler, and Notifier (see Chapters 9-10); (b) configure the Core Component using the Platform Administrator's Guide (see Tables C-1 and C-7 of that guide).

This procedure ends here.

**Installing the Core Component on WebLogic**

**Prerequisites**
- All General Prerequisites (page 71) are satisfied.

**Action**

To install the Core Component on WebLogic:

1. On the Application Server host, create a directory that will serve as the OneSpan Sign home directory.

2. Into this [eSL_HOME] directory, copy the file silanisLicense.xml provided by OneSpan.

3. Configure the Application Server with the following Java Virtual Machine (JVM) system properties:
   - AWS_HOME = [eSL_HOME]
   - RESOURCE_SERVER_LOCATION = https://<Resource Manager Server>:<port>/resources-master/storage
4. If you want to change the default expiration time of the local cache (which is 900 seconds), configure the Application Server with the following Java Virtual Machine (JVM) system property:
   
   • `CACHE_LIFETIME_SECONDS = <N>`
   
   **Step 5 and Step 6 are required if you are using vertical clustering, which is not recommended.**

5. If you want to change the local cache location (which by default is the system-dependent temporary-file directory, together with the suffix `esp`), configure the Application Server with the following Java Virtual Machine (JVM) system property:
   
   • `RESOURCE_LOCAL_CACHE = <path to local cache>`

   The directory of this local cache should never be auto-cleaned. Yet on RHEL/CentOS, the system-dependent temporary-file directory is auto-cleaned by default.

6. The Core Component submits frequent requests to the PDF Document Engine. Each request has an ID that looks something like this: `fc70e50d-977b-48d4-8e60-b3474daa0028-2984-<string>`. Here `<string>` is a character string whose default value is the host name of the machine where the installation is taking place. If multiple Core Component instances are installed on that machine, using that default value may be confusing because no ID can uniquely identify its associated instance. If you want to avoid that confusion, configure the Application Server with the following JVM system property:
   
   • `aws.jvm.instance.name = <Core-Component-instance name>`

   For example, if you set that property to "awsinstancenode1", a typical ID will look something like this: `bbeea390-3687-4b69-a8e3-b17e9608c2c2-awsinstancenode1`.

7. In the `[esl_HOME]` directory, create one directory called "logs" and another directory called "logging".

8. Configure the Apache log4j logging facility by doing the following:
   
   a. Copy the following files to the Application Server's `classpath` in the directory `<domain_name>/lib:` (1) the file `log4j.jar` from the "Server/Core Components/Thirdparty" directory of the software provided by OneSpan; (2) the file `wllog4j.jar` provided by Oracle WebLogic.
   
   b. Into the "logging" directory, copy the file "log4j.properties" from the "Server/Core Components/Resource" directory of the software provided by OneSpan.
c. Edit the "log4j.properties" file you just copied, configuring it appropriately for your needs.

For detailed information about the Apache log4j logging facility, go to:

- http://logging.apache.org/log4j/1.2/

d. Configure the Application Server with the following JVM system properties:

- log4j.configuration = file:[$ESL_HOME]/logging/log4j.properties
- weblogic.log.Log4jLoggingEnabled = true
- SIL_LOG_FRAMEWORK = LOG4J

9. If you are installing the Core Component in a clustered Application Server environment, do one of the following:

- Copy the OneSpan Sign home directory created in the preceding steps to each node in the cluster.
- Ensure that the OneSpan Sign home directory is on a network drive that can be accessed by all nodes in the cluster.

10. Create a Data Source for the Core Database using: (i) the parameters in Table 7-1; (ii) the appropriate procedure from Creating a JDBC Provider (page 316).

The user account for the Data Source should have read/write access to the Core Database.

11. If any of your e-signature processes must fetch documents from a database (instead of from the File System), create a Data Source for the Document Retriever Pool Database using: (i) the parameters in Table 8-5; (ii) the appropriate procedure from Creating a JDBC Provider (page 316).

The user account for the Data Source should have READ access to the Document Retriever Pool Database.

12. Create JMS resources for the Core Component by doing the following:

a. Create a Queue Connection Factory with JNDI name "jms/QueueConnectionFactory".

b. Create the following persistent JMS queues:

- ErrorQueue (use the parameters in Table 8-1)
- PostOnlineQueue (use the parameters in Table 8-2)
- ConcatenateDocumentsQueue (use the parameters in Table 8-3)
- NotifierResponseQueue (use the parameters in Table 8-4)
13. Inside the Application Server, deploy the "eSL-Core.ear" file provided by OneSpan.

14. If you are using "out-of-process events" (i.e., general system events that are independent of any particular e-signature process), configure the following Java system parameter (default = false):
   -Dinclude.outofprocess.event = true

   This parameter is a flag that determines if out-of-process events will be included in e-signature processes.

15. If you have a remote connection to the JMS queue of the Scheduler and/or if you want to change the Queue Connection Factory name from that specified in Step 12 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "schedulerJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator’s Guide).

16. If you have a remote connection to the JMS queue of the Notifier and/or if you want to change the Queue Connection Factory name from that specified in Step 12 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "notifierJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator’s Guide).

17. Configure HTTPS access to the Core Component.

18. Save and enable your changes.

19. Restart the Application Server.

20. Verify the above installation by going to the following URL:
   - https://<server>:<port>/esep/rest/application.wadl

   Here <server> denotes the server where the Core Component is installed, and <port> denotes the port it uses. You should be able to see this file’s contents, or be able to download the file.

   To enable the Core Component to function properly, you must also:
   (a) install the Event Manager, Scheduler, and Notifier (see Chapters 9-10); (b) configure the Core Component using the Platform Administrator’s Guide (see Tables C-1 and C-7 of that guide).

This procedure ends here.

**Uninstalling the Core Component**

To uninstall the Core Component, you need perform only one of the following procedures (depending on your Application Server):

- Uninstalling the Core Component on WebSphere (page 85)
- Uninstalling the Core Component on JBoss or Wildfly (page 86)
- Uninstalling the Core Component on WebLogic (page 87)
Uninstalling the Core Component on WebSphere

This section describes how to manually uninstall the Core Component on WebSphere. Chapter 3 describes an alternate, automated way of doing so.

To uninstall the Core Component on WebSphere:

1. Inside the Application Server, uninstall the Core Component named "eSL-Core" as well as the "eSL-Resources Manager" component.
2. Delete the activation specifications created when the Core Component was installed.
3. Delete the JMS queues created when the Core Component was installed.

Perform the next step only if you are **not using the e-Vault Manager**. The e-Vault Manager employs the Queue Connection Factory mentioned in the next step, so deleting it would harm the e-Vault Manager.

4. Delete the Queue Connection Factory created when the Core Component was installed.
5. If you used WebSphere SIB default messaging:
   a. Remove the bus members and destination created when the Core Component was installed.
   b. Delete the bus created when the Core Component was installed.

6. If you used WebSphere MQ messaging, delete the queues created on the MQ server when the Core Component was installed.
7. Delete the Data Source for the Core Database. If you created a Data Source for the Document Retriever Pool Database, delete it as well.
8. Delete the JDBC Provider for the Core Database (provided it’s not being used by any other application). If applicable, do the same for the Document Retriever Pool Database.
9. If you copied any JDBC driver files for the Core Database or Document Retriever Pool Database when the Core Component was installed, delete them (provided they’re not being used by any other application).
10. Unset the generic JVM arguments that were set when the Core Component was installed.
11. Perform the following related steps (if you are using a cluster, perform these steps on each Application Server host):
   a. Make a backup copy of the **OneSpan Sign** home directory.

   If you do not have a backup copy of this directory, you will permanently lose all its data.

   b. Delete the **OneSpan Sign** home directory and all its contents.
12. Perform the following related steps:
   a. Make a backup copy of the Core Database.

   ![Warning]
   If you do not have a backup copy of this database, you will permanently lose all its data.

   b. Delete the Core Database and all its contents.

13. Restart the Application Server.

14. Sequentially perform the following procedures:
   a. Uninstalling the Notifier (page 109)
   b. Uninstalling the Scheduler (page 107)
   c. Uninstalling the Event Manager (page 92)

   This procedure ends here.

**Uninstalling the Core Component on JBoss or Wildfly**

This section describes how to manually uninstall the Core Component on JBoss or Wildfly. Chapter 3 describes an alternate, automated way of doing so.

**To uninstall the Core Component on JBoss or Wildfly:**

1. Inside the Application Server, delete the file "eSL-Core.ear".

2. Delete relevant JMS resources by doing the following:
   a. Delete the JMS queues created when the Core Component was installed.

   ![Warning]
   Perform the next step only if you are not using the e-Vault Manager™. The e-Vault Manager™ employs the Queue Connection Factory mentioned in the next step, so deleting it would harm the e-Vault Manager™.

   b. Delete the Queue Connection Factory created when the Core Component was installed.

3. Delete the Data Sources for the Core Database and the Document Retriever Pool Database.

4. Delete the JDBC Providers for the Core Database and the Document Retriever Pool Database (provided they’re not being used by any other application).

5. If you copied any JDBC driver files for the Core Database or Document Retriever Pool Database when the Core Component was installed, delete them (provided they’re not being used by any other application).

6. Unset the generic JVM arguments that were set when the Core Component was installed.
7. Perform the following related steps on the Application Server host:
   a. Make a backup copy of the OneSpan Sign home directory.

   ![Warning]
   If you do not have a backup copy of this directory, you will permanently lose all its data.

   b. Delete the OneSpan Sign home directory and all its contents.

8. Perform the following related steps:
   a. Make a backup copy of the Core Database.

   ![Warning]
   If you do not have a backup copy of this database, you will permanently lose all its data.

   b. Delete the Core Database and all its contents.

9. Restart the Application Server.

10. Sequentially perform the following procedures:
    a. Uninstalling the Notifier (page 109)
    b. Uninstalling the Scheduler (page 107)
    c. Uninstalling the Event Manager (page 92)

This procedure ends here.

Uninstalling the Core Component on WebLogic

To uninstall the Core Component on WebLogic:

1. Inside the Application Server, uninstall the Core Component named "eSL-Core" as well as the "eSL-Resources Manager" component.

2. Delete the JMS queues created when the Core Component was installed.

   ![Warning]
   Perform the next step only if you are not using the e-Vault Manager™. The e-Vault Manager™ employs the Queue Connection Factory mentioned in the next step, so deleting it would harm the e-Vault Manager™.

3. Delete the Queue Connection Factory created when the Core Component was installed.

4. Delete the Data Source for the Core Database. If you created a Data Source for the Document Retriever Pool Database, delete it as well.

5. Delete the JDBC Provider for the Core Database (provided it’s not being used by any other application). If applicable, do the same for the Document Retriever Pool Database.

6. If you were not using an Oracle database when the Core Component was installed, and if you copied any JDBC driver files for the Core
Database or the Document Retriever Pool Database at that time, delete them (provided they’re not being used by any other application).

7. Unset the generic JVM arguments that were set when the Core Component was installed.

8. Perform the following related steps (if you are using a cluster, perform these steps on each Application Server host):
   a. Make a backup copy of the OneSpan Sign home directory.

   ![Warning]
   If you do not have a backup copy of this directory, you will permanently lose all its data.

   b. Delete the OneSpan Sign home directory and all its contents.

9. Perform the following related steps:
   a. Make a backup copy of the Core Database.

   ![Warning]
   If you do not have a backup copy of this database, you will permanently lose all its data.

   b. Delete the Core Database and all its contents.

10. Restart the Application Server.

11. Sequentially perform the following procedures:
   a. Uninstalling the Notifier (page 109)
   b. Uninstalling the Scheduler (page 107)
   c. Uninstalling the Event Manager (page 92)

   This procedure ends here.
CHAPTER 9: Deploying the Event Manager

The Event Manager is a OneSpan Sign module that monitors internal events. This feature was introduced in OneSpan Sign 5.1.

This chapter describes:

- Installing the Event Manager (page 89)
- Uninstalling the Event Manager (page 92)

The Event Manager is required by the Core Component.

Installing the Event Manager

To install the Event Manager, you need perform only one of the following procedures (depending on your Application Server):

- Installing the Event Manager on WebSphere (page 89)
- Installing the Event Manager on JBoss or Wildfly (page 91)
- Installing the Event Manager on WebLogic (page 91)

Installing the Event Manager on WebSphere

This section describes how to manually install the Event Manager on WebSphere. Chapter 3 describes an alternate, automated way of doing so.

Prerequisites

- Your environment meets all relevant requirements in Chapter 2.
- You have performed the relevant procedure from Preparing an App Server Environment (page 312).
- The procedure Installing the Admin Console (page 65) has been performed.
- The Event Manager must be installed on the same Application Server instances as the Core Component.

Action

To install the Event Manager on WebSphere:

1. If you are using WebSphere SIB default messaging, create the following destination within the bus you created for the Core Component in Chapter 8:
   - EventsQueueDestination
2. To complete the configuration of JMS resources:
   a. Using the parameters in Table 9-1, create a persistent JMS queue called EventsQueue.
      
      For the above queue: (1) from the WebSphere Administration Console, go to Resources > JMS >
      Queues > Queue name > Advanced properties; (2) select the checkbox **Append RFH version 2
      headers to messages sent to this destination** Message.

b. Create the following activation specification:
   - Name: ActivationSpecEvents
   - JNDI name: jms/ActivationSpecEvents
   - Destination JNDI name: jms/queue/EventsQueue

3. Inside the Application Server, deploy the file "eSL-Events.ear" provided by OneSpan.

4. Restart the Application Server.

5. Verify the installation by doing the following:
   a. Start the Application Server software, and verify that during
      startup no errors were logged in the OneSpan Sign log or the
      Application Server log.

   b. Use the Application Server software to verify that the Event
      Manager was properly deployed.

   This procedure ends here.

### Table 9-1: Event Manager Queue parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Events Queue</td>
<td>EventsQueue</td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
<td>APPROVEIT.EVENTS.QUEUE</td>
</tr>
<tr>
<td>JNDI name associated with the Events Queue</td>
<td>The JNDI name should be</td>
</tr>
<tr>
<td></td>
<td>&quot;jms/queue/EventsQueue&quot;</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
<td>This value must be <strong>Persistent</strong>.</td>
</tr>
<tr>
<td></td>
<td>Default = Application</td>
</tr>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
<td>This value must be &quot;5&quot;.</td>
</tr>
<tr>
<td></td>
<td>Default = 2</td>
</tr>
</tbody>
</table>
Installing the Event Manager on JBoss or Wildfly

This section describes how to manually install the Event Manager on JBoss or Wildfly. Chapter 3 describes an alternate, automated way of doing so.

Prerequisites

- Your environment meets all relevant requirements in Chapter 2.
- You have performed the relevant procedure from Preparing an App Server Environment (page 312).
- The procedure Installing the Admin Console (page 65) has been performed.
- The Event Manager must be installed on the same Application Server instances as the Core Component.

Action

To install the Event Manager on JBoss or Wildfly:

1. Create a persistent JMS queue called EventsQueue using the parameters in Table 9-1.
2. Inside the Application Server, deploy the file "eSL-Events.ear".
3. Restart the Application Server.
4. Verify the installation by doing the following:
   a. Start the Application Server software, and verify that during startup no errors were logged in the OneSpan Sign log or the Application Server log.
   b. Use the Application Server software to verify that the Event Manager was properly deployed.

This procedure ends here.

Installing the Event Manager on WebLogic

Prerequisites

- Your environment meets all relevant requirements in Chapter 2.
- You have performed the relevant procedure from Preparing an App Server Environment (page 312).
- The procedure Installing the Admin Console (page 133) has been performed.
- The Event Manager must be installed on the same Application Server instances as the Core Component.

Action

To install the Event Manager on WebLogic:

1. Using the parameters in Table 9-1, create a persistent JMS queue
Uninstalling the Event Manager
called EventsQueue.
2. Inside the Application Server, deploy the file "eSL-Events.ear"
   provided by OneSpan.
3. Restart the Application Server.
4. Verify the installation by doing the following:
   a. Start the Application Server software, and verify that during
      startup no errors were logged in the OneSpan Sign log or the
      Application Server log.
   b. Use the Application Server software to verify that the Event
      Manager was properly deployed.

   This procedure ends here.

Uninstalling the Event Manager

To uninstall the Event Manager, you need perform only one of the following
procedures (depending on your Application Server):

- Uninstalling the Event Manager on WebSphere (page 92)
- Uninstalling the Event Manager on JBoss or Wildfly (page 93)
- Uninstalling the Event Manager on WebLogic (page 93)

Uninstalling the Event Manager on WebSphere

This section describes how to *manually* uninstall the Event Manager on
WebSphere. Chapter 3 describes an alternate, *automated* way of doing so.

To uninstall the Event Manager on WebSphere:

1. Inside the Application Server, uninstall the "eSL Event Manager"
   application.
2. Delete the activation specification created during the installation of
   the Event Manager.
3. Delete the JMS queue EventsQueue.
4. If you used WebSphere SIB default messaging, remove the bus
   destination EventsQueueDestination.
5. If you used WebSphere MQ messaging, on the MQ server delete the
   queue APPROVEIT.EVENTS.QUEUE.
6. Delete the directory in which JMS queue messages were to be stored.
7. Restart the Application Server.

   This procedure ends here.
Uninstalling the Event Manager on JBoss or Wildfly

This section describes how to manually uninstall the Event Manager on JBoss or Wildfly. Chapter 3 describes an alternate, automated way of doing so.

To uninstall the Event Manager on JBoss or Wildfly:

1. Inside the Application Server, delete the file "eSL-Events.ear".
2. Delete the JMS queue EventsQueue.
3. Delete the directory in which JMS queue messages were to be stored.
4. Restart the Application Server.

   This procedure ends here.

Uninstalling the Event Manager on WebLogic

To uninstall the Event Manager on WebLogic:

1. Inside the Application Server, uninstall the "eSL Event Manager" application.
2. Delete the JMS queue EventsQueue.
3. Delete the directory in which JMS queue messages were to be stored.
4. Restart the Application Server.

   This procedure ends here.
CHAPTER 10: Deploying the e-Mail Notification Manager

The e-Mail Notification Manager is a OneSpan Sign component that invites users by e-Mail to sign documents. In addition to the initial invitation, the e-Mail Notification Manager can also send periodic reminders until an expiration date is reached. If OneSpan Sign determines that a user has accepted an invitation, it will ensure that no more reminders are sent.

The e-Mail Notification Manager consists of two sub-components:

- **Scheduler** — This sub-component schedules a notification to be sent at a future time.
- **Notifier** — This sub-component sends notifications. Important examples of such notifications are e-Mail warnings sent to the OneSpan Sign Administrator about the PDF Document Engine (e.g., *A certificate is about to expire*).

To install the e-Mail Notification Manager™, you must separately install the Scheduler and then the Notifier, as described in this chapter.

This chapter describes:

- **Installing the Scheduler (page 94)**
- **Installing the Notifier (page 102)**
- **Uninstalling the Scheduler (page 107)**
- **Uninstalling the Notifier (page 109)**

The Scheduler and Notifier are required by the Core Component.

**Installing the Scheduler**

To install the Scheduler, you need perform only one of the following procedures (depending on your Application Server):

- **Installing the Scheduler on WebSphere (page 94)**
- **Installing the Scheduler on JBoss or Wildfly (page 98)**
- **Installing the Scheduler on WebLogic (page 100)**

**Installing the Scheduler on WebSphere**

This section describes how to manually install the Scheduler on WebSphere. Chapter 3 describes an alternate, automated way of doing so.

**Prerequisites**

- Your environment meets all relevant requirements in Chapter 2.
- You have performed the relevant procedure from Preparing an App.
Server Environment (page 312).

- The following procedures have been performed:
  - Creating the e-Mail Notification Manager Database (page 283)
  - Installing the Admin Console (page 65)
  - Installing the Scheduler on WebSphere (page 312).

- If you have a remote connection to the JMS queue of the Core Component and/or the Notifier and/or if you want to change the Queue Connection Factory name from that specified in Chapter 3 or Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory:
  - If you are using WebSphere SIB default messaging, you have created a bus by performing Step 13a of Installing the Core Component on WebSphere (page 72).
  - If you are using WebSphere, you have created a Queue Connection Factory by performing Step 14a of Installing the Core Component on WebSphere (page 72).

The Scheduler should be installed on the same Application Server instances as the Core Component.

Action

To install the Scheduler on WebSphere:

1. If you are installing the Scheduler in a clustered Application Server environment, prevent database deadlocks by using the Admin Console to edit the file "silanis_quartz.properties" (see Table B-4 of the Platform Administrator's Guide). In particular, do the following:
   a. Set the value of the parameter org.quartz.jobStore.isClustered to true.
   b. Add a parameter called org.quartz.scheduler.instanceId, and set its value to AUTO.

   If you are preparing a clustered environment, you should use a suitable time-sync service to synchronize the clocks on all cluster members to within one second of each other. For more on this, see:
   http://www.boulder.nist.gov/timefreq/service/its.htm

   For more on configuring the Quartz Scheduler, see:
   http://www.quartz-scheduler.org/documentation/quartz-2.x/configuration/index

2. Create a Data Source for the e-Mail Notification Manager Database, using: (i) the parameters in Table 10-1; (ii) the appropriate procedure from Creating a JDBC Provider (page 316). Ensure that Container Managed Persistence (CMP) is not selected.

   The user account for the Data Source should have read/write access to the e-Mail Notification Manager Database.
3. Configure the Application Server with the following Java Virtual Machine (JVM) system property:
   • RESOURCE_SERVER_LOCATION = https://<Resource Manager Server>:<port>/resources-master/storage

4. If you are using WebSphere SIB default messaging, create the following destination within the bus you created for the Core Component in Chapter 8:
   • SchedulerQueueDestination

5. To complete the configuration of JMS resources:
   a. Using the parameters in Table 10-2, create a persistent JMS queue called SchedulerRequestQueue.

   For the above queue: (1) from the WebSphere Administration Console, go to Resources > JMS > Queues > Queue name > Advanced properties; (2) select the checkbox Append RFH version 2 headers to messages sent to this destination Message.

   b. Create the following activation specification:
      • Name: ActivationSpecScheduler
      • JNDI name: jms/ActivationSpecScheduler
      • Destination JNDI name: jms/queue/SchedulerRequestQueue

6. Use the Admin Console to configure the file "silanis_quartz.properties" by setting "org.quartz.dataSource.SchedulerDS.jndiURL" to the JNDI URL of the Data Source created in Step 2. For a description of that file, see: http://www.quartz-scheduler.org/documentation/quartz-2.x/configuration/index.

7. If you are using an Oracle Database, and if the tables created in Creating the e-Mail Notification Manager Database (page 283) (see the Prerequisites) are not under the same schema as the user of the Data Source created in Step 2, use the Admin Console to change the "silanis_quartz.properties" file by setting:

   org.quartz.jobStore.tablePrefix = <schema>.QRTZ_

   Here <schema> is the name of the schema under which e-Mail Notification Manager™ tables were created in Creating the e-Mail Notification Manager Database (page 283) (see the Prerequisites).

8. Inside the Application Server, deploy the "eSL-Scheduler.ear" file provided by OneSpan.

9. If you have a remote connection to the JMS queue of the Core Component and/or if you want to change the Queue Connection Factory name from that specified in Chapter 3 or Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "awsJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator’s Guide).

10. If you have a remote connection to the JMS queue of the Notifier and/or if you want to change the Queue Connection Factory name from that
specified in Chapter 3 or Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "notifierJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator’s Guide).

11. Restart the Application Server.

12. Verify the installation by doing the following:

   a. Start the Application Server software, and verify that during startup no errors were logged in the e-Mail Notification Manager™ log or the Application Server log.

   b. Use the Application Server software to verify that the Scheduler was properly deployed.

13. Optional: If you want to change the logging behavior of the e-Mail Notification Manager™, consult Appendix B.

⚠️ You must also configure the Scheduler using the Platform Administrator’s Guide (see Tables B-4 and C-7 of that guide).

This procedure ends here.

**Table 10-1: e-Mail Notification Manager’s Data Source settings**

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the e-Mail Notification Manager™ Data Source</td>
<td>Any valid character string</td>
</tr>
<tr>
<td>JNDI Name associated with the e-Mail Notification Manager™ Data Source</td>
<td>This name must be &quot;scheduler/jdbc/ConnectionPool&quot;.</td>
</tr>
<tr>
<td>Transaction Isolation Level</td>
<td>READ_COMMITTED</td>
</tr>
<tr>
<td>Definition of the JBoss/Wildfly driver referenced by the e-Mail Notification Manager™ Data Source</td>
<td>Possible values:</td>
</tr>
<tr>
<td></td>
<td>• oracle.ojdbc (Oracle)</td>
</tr>
<tr>
<td></td>
<td>• ibm.db2 (DB2)</td>
</tr>
<tr>
<td></td>
<td>• com.microsoft (MS SQL Server)</td>
</tr>
<tr>
<td></td>
<td>• mysql (MySQL)</td>
</tr>
</tbody>
</table>

**Table 10-2: Scheduler Queue parameters**

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Notification Queue</td>
<td>SchedulerRequestQueue</td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
<td>APPROVEIT.SCHEDULER.QUEUE</td>
</tr>
<tr>
<td>JNDI name associated with the Notification Queue</td>
<td>The JNDI name should be &quot;jms/queue/SchedulerRequestQueue&quot;.</td>
</tr>
</tbody>
</table>
Installing the Scheduler on JBoss or Wildfly

Prerequisites

- Your environment meets all relevant requirements in Chapter 2.
- You have performed the relevant procedure from Preparing an App Server Environment (page 312).
- The following procedures have been performed:
  - Creating the e-Mail Notification Manager Database (page 283)
  - Installing the Admin Console (page 65)
- If you have a remote connection to the JMS queue of the Core Component and/or the Notifier and/or if you want to change the Queue Connection Factory name from that specified in Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory:
  - You have created a Queue Connection Factory by performing Step 12a of Installing the Core Component on JBoss or Wildfly (page 79).

Action

To install the Scheduler on JBoss or Wildfly:

1. Create a Data Source for the e-Mail Notification Manager™ Database using: (i) the parameters in Table 10-1; (ii) the appropriate procedure from Creating a JDBC Provider (page 316).

   The user account for the Data Source should have read/write access to the e-Mail Notification Manager™ Database.

2. Configure the Application Server with the following Java Virtual Machine (JVM) system property:
   - RESOURCE_SERVER_LOCATION = https://<Resource Manager Server>:<port>/resources-master/storage
3. Create JMS resources for the Scheduler by doing the following:
   a. Using the parameters in Table 10-2, create a persistent JMS queue called
      SchedulerRequestQueue.

4. Use the Admin Console to configure the file "silanis_quartz.properties" by setting "org.quartz.dataSource.SchedulerDS.jndiURL" to the JNDI URL of the Data Source created in Step 1. For a description of that file, see: http://www.quartz-scheduler.org/documentation/quartz-2.x/configuration/index.

5. If you are using an Oracle Database, and if the tables created in Creating the e-Mail Notification Manager Database (page 283) (see the Prerequisites) are not under the same schema as the user of the Data Source created in Step 1, use the Admin Console to change the "silanis_quartz.properties" file by setting:

   org.quartz.jobStore.tablePrefix = <schema>.QRTZ_<schema>

   Here <schema> is the name of the schema under which e-Mail Notification Manager™ tables were created in Creating the e-Mail Notification Manager Database (page 283) (see the Prerequisites).

6. Inside the Application Server, deploy the "eSL-Scheduler.ear" file provided by OneSpan.

7. If you have a remote connection to the JMS queue of the Core Component and/or if you want to change the Queue Connection Factory name from that specified in Chapter 3 or Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "awsJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator’s Guide).

8. If you have a remote connection to the JMS queue of the Notifier and/or if you want to change the Queue Connection Factory name from that specified in Chapter 3 or Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "notifierJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator’s Guide).

9. Restart the Application Server.

10. Verify the installation by doing the following:
    a. Start the Application Server software, and verify that during startup no errors were logged in the e-Mail Notification Manager™ log or the Application Server log.
    b. Use the Application Server software to verify that the Scheduler was properly deployed.

11. Optional: If you want to change the logging behavior of the e-Mail Notification Manager™, consult Appendix B.

You must also configure the Scheduler using the Platform Administrator’s Guide (see Tables B-4 and C-7 of that guide).

This procedure ends here.
Installing the Scheduler on WebLogic

Prerequisites

- Your environment meets all relevant requirements in Chapter 2.
- You have performed the relevant procedure from Preparing an App Server Environment (page 312).
- The following procedures have been performed:
  - Creating the e-Mail Notification Manager Database (page 283)
  - Installing the Admin Console (page 65)
- If you have a remote connection to the JMS queue of the Core Component and/or the Notifier and/or if you want to change the Queue Connection Factory name from that specified in Chapter 3 or Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory:
  - You have created a Queue Connection Factory by performing Step 13a of Installing the Core Component on WebLogic (page 81).

⚠️ The Scheduler should be installed on the same Application Server instances as the Core Component.

Action

To install the Scheduler on WebLogic:

1. If you are installing the Scheduler in a clustered Application Server environment, prevent database deadlocks by using the Admin Console to edit the file "silanis_quartz.properties" (see Table B-4 of the Platform Administrator's Guide). In particular, do the following:
   a. Set the value of the parameter org.quartz.jobStore.isClustered to true.
   b. Add a parameter called org.quartz.scheduler.instanceId, and set its value to AUTO.

⚠️ If you are preparing a clustered environment, you should use a suitable time-sync service to synchronize the clocks on all cluster members to within one second of each other. For more on this, see: http://www.boulder.nist.gov/timefreq/service/its.htm

For more on configuring the Quartz Scheduler, see: http://www.quartz-scheduler.org/documentation/quartz-2.x/configuration/index

2. Create a Data Source for the e-Mail Notification Manager Database, using: (i) the parameters in Table 10-1; (ii) the appropriate procedure from Creating a JDBC Provider (page 316).

⚠️ The user account for the Data Source should have read/write access to the e-Mail Notification Manager Database.
3. Configure the Application Server with the following Java Virtual Machine (JVM) system property:
   • RESOURCE_SERVER_LOCATION = https://<ResourceManager Server>:<port>/resources-master/storage

4. Using the parameters in Table 10-2, create a persistent JMS queue called SchedulerRequestQueue.

5. Use the Admin Console to configure the file "silanis_quartz.properties" by setting "org.quartz.dataSource.SchedulerDS.jndiURL" to the JNDI URL of the Data Source created in Step 2. For a description of that file, see: http://www.quartz-scheduler.org/documentation/quartz-2.x/configuration/index.

6. If you are using an Oracle Database, and if the tables created in Creating the e-Mail Notification Manager Database (page 283) (see the Prerequisites) are not under the same schema as the user of the Data Source created in Step 2, use the Admin Console to change the "silanis_quartz.properties" file by setting:

   org.quartz.jobStore.tablePrefix = <schema>.QRTZ_

   Here <schema> is the name of the schema under which e-Mail Notification Manager™ tables were created in Creating the e-Mail Notification Manager Database (page 283) (see the Prerequisites).

7. Inside the Application Server, deploy the "eSL-Scheduler.ear" file provided by OneSpan.

8. If you have a remote connection to the JMS queue of the Core Component and/or if you want to change the Queue Connection Factory name from that specified in Chapter 3 or Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "awsJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator’s Guide).

9. If you have a remote connection to the JMS queue of the Notifier and/or if you want to change the Queue Connection Factory name from that specified in Chapter 3 or Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "notifierJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator’s Guide).

10. Restart the Application Server.

11. Verify the installation by doing the following:
   a. Start the Application Server software, and verify that during startup no errors were logged in the e-Mail Notification Manager™ log or the Application Server log.
   b. Use the Application Server software to verify that the Scheduler was properly deployed.
12. **Optional:** If you want to change the logging behavior of the e-Mail Notification Manager™, consult Appendix B.

You must also configure the Scheduler using the *Platform Administrator’s Guide* (see Tables B-4 and C-7 of that guide).

This procedure ends here.

### Installing the Notifier

To install the Notifier, you need perform only one of the following procedures (depending on your Application Server):

- Installing the Notifier on WebSphere (page 102)
- Installing the Notifier on JBoss or Wildfly (page 104)
- Installing the Notifier on WebLogic (page 105)

### Installing the Notifier on WebSphere

This section describes how to *manually* install the Notifier on WebSphere. Chapter 3 describes an alternate, *automated* way of doing so.

**Prerequisites**

- Your environment meets all relevant requirements in Chapter 2.
- You have performed the relevant procedure from Preparing an App Server Environment (page 312).
- If you have a remote connection to the JMS queue of the Core Component and/or the Scheduler and/or if you want to change the Queue Connection Factory name from that specified in Chapter 3 or Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory:
  - If you are using WebSphere SIB default messaging, you have created a bus by performing Step 13a of Installing the Core Component on WebSphere (page 72).
  - If you are using WebSphere, you have created a Queue Connection Factory by performing Step 14a of Installing the Core Component on WebSphere (page 72).

The Notifier should be installed on the same Application Server instances as the Core Component.

**Action**

**To install the Notifier on WebSphere:**

1. If you are using WebSphere SIB default messaging, create the following destination within the bus you created for the Core Component in Chapter 8.
2. Configure the Application Server with the following Java Virtual Machine (JVM) system property:
   - 
     \[
     \text{RESOURCE\_SERVER\_LOCATION} = \text{https://<Resource\_Manager\_Server>:<port>/resources-master/storage}
     \]

3. To complete the configuration of JMS resources:
   a. Using the parameters in Table 10-3, create a persistent JMS queue called `NotifierRequestQueue`.

   For the above queue: (1) from the WebSphere Administration Console, go to Resources > JMS > Queues > Queue name > Advanced properties; (2) select the checkbox **Append RFH version 2 headers to messages sent to this destination Message.**

   b. Create the following activation specification:
      - Name: ActivationSpecNotification
      - JNDI name: jms/ActivationSpecNotificationRequest
      - Destination JNDI name: jms/queue/NotifierRequestQueue

4. If you are using an authenticated SMTP connection, configure "mailserverconfig.properties" using the Admin Console (see Table B-3 of the Platform Administrator’s Guide).

5. Inside the Application Server, deploy the "eSL-Notifier.ear" file provided by OneSpan.

6. If you have a remote connection to the JMS queue of the Core Component and/or if you want to change the Queue Connection Factory name from that specified in Chapter 3 or Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "awsJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator’s Guide).

7. If you have a remote connection to the JMS queue of the Scheduler and/or if you want to change the Queue Connection Factory name from that specified in Chapter 3 or Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "schedulerJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator’s Guide).

8. Restart the Application Server.

9. Verify the installation by doing the following:
   a. Start the Application Server software, and verify that during startup no errors were logged in the e-Mail Notification Manager™ log or the Application Server log.
   b. Use the Application Server software to verify that the Notifier was properly deployed.
10. **Optional:** If you want to change the logging behavior of the e-Mail Notification Manager™, consult Appendix B.

⚠️ You must also configure the Notifier using the *Platform Administrator’s Guide* (see Tables B-3 and C-7 of that guide).

This procedure ends here.

<table>
<thead>
<tr>
<th>Table 10-3: Notifier Queue parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DESCRIPTION</strong></td>
</tr>
<tr>
<td>Name of the Notification Queue</td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
</tr>
<tr>
<td>JNDI name associated with the Notification Queue</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
</tr>
<tr>
<td>Default = Application</td>
</tr>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
</tr>
</tbody>
</table>

---

**Installing the Notifier on JBoss or Wildfly**

This section describes how to *manually* install the Notifier on JBoss or Wildfly. Chapter 3 describes an alternate, *automated* way of doing so.

**Prerequisites**

- Your environment meets all relevant requirements in *Chapter 2*.
- You have performed the relevant procedure from *Preparing an App Server Environment* (page 312).
- If you have a remote connection to the JMS queue of the Core Component and/or the Scheduler and/or if you want to change the Queue Connection Factory name from that specified in *Chapter 8* and/or if you want to use credentials for the queues and the Queue Connection Factory:
  - You have created a Queue Connection Factory by performing *Step 12a* of *Installing the Core Component on JBoss or Wildfly* (page 79).

⚠️ The Notifier should be installed on the same Application Server instances as the Core Component.

**Action**

To install the Notifier on JBoss or Wildfly:

1. Create a persistent JMS queue called *NotifierRequestQueue* using the
parameters in Table 10-3.

2. Configure the Application Server with the following Java Virtual Machine (JVM) system property:
   - RESOURCE_SERVER_LOCATION = https://<Resource Manager Server>:<port>/resources-master/storage

3. If you are using an authenticated SMTP connection, configure "mailserverconfig.properties" using the Admin Console (see Table B-3 of the Platform Administrator's Guide).

4. Inside the Application Server, deploy the "eSL-Notifier.ear" file provided by OneSpan.

5. If you have a remote connection to the JMS queue of the Core Component and/or if you want to change the Queue Connection Factory name from that specified in Chapter 3 or Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "awsJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator's Guide).

6. If you have a remote connection to the JMS queue of the Scheduler and/or if you want to change the Queue Connection Factory name from that specified in Chapter 3 or Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "schedulerJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator's Guide).

7. Restart the Application Server.

8. Verify the installation by doing the following:
   a. Start the Application Server software, and verify that during startup no errors were logged in the e-Mail Notification Manager log or the Application Server log.
   b. Use the Application Server software to verify that the Notifier was properly deployed.

9. Optional: If you want to change the logging behavior of the e-Mail Notification Manager, consult Appendix B.

You must also configure the Notifier using the Platform Administrator's Guide (see Table C-7 of that guide).

This procedure ends here.

**Installing the Notifier on WebLogic**

**Prerequisites**

- Your environment meets all relevant requirements in Chapter 2.
- You have performed the relevant procedure from Preparing an App Server Environment (page 312).
- If you have a remote connection to the JMS queue of the Core Component and/or the Scheduler and/or if you want to change the Queue Connection Factory.
Factory name from that specified in Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory:

- You have created a Queue Connection Factory by performing Step 13a of Installing the Core Component on WebLogic (page 81).

The Notifier should be installed on the same Application Server instances as the Core Component.

**Action**

**To install the Notifier on WebLogic:**

1. Create a persistent JMS queue called NotifierRequestQueue using the parameters in Table 10-3.

2. Configure the Application Server with the following Java Virtual Machine (JVM) system property:
   - RESOURCE_SERVER_LOCATION = https://<Resource Manager Server>:<port>/resources-master/storage

3. If you are using an authenticated SMTP connection, configure "mailserverconfig.properties" using the Admin Console (see Table B-3 of the Platform Administrator’s Guide).

4. Inside the Application Server, deploy the "eSL-Notifier.ear" file provided by OneSpan.

5. If you have a remote connection to the JMS queue of the Core Component and/or if you want to change the Queue Connection Factory name from that specified in Chapter 3 or Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "awsJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator’s Guide).

6. If you have a remote connection to the JMS queue of the Scheduler and/or if you want to change the Queue Connection Factory name from that specified in Chapter 3 or Chapter 8 and/or if you want to use credentials for the queues and the Queue Connection Factory, configure "schedulerJMS.properties" using the Admin Console (see Table C-7 of the Platform Administrator’s Guide).

7. Restart the Application Server.

8. Verify the installation by doing the following:
   a. Start the Application Server software, and verify that during startup no errors were logged in the e-Mail Notification Manager™ log or the Application Server log.
   b. Use the Application Server software to verify that the Notifier was properly deployed.
9. Optional: If you want to change the logging behavior of the e-Mail Notification Manager™, consult Appendix B.

You must also configure the Notifier using the Platform Administrator’s Guide (see Table C-7 of that guide).

This procedure ends here.

**Uninstalling the Scheduler**

To uninstall the Scheduler, you need perform only one of the following procedures (depending on your Application Server):

- Uninstalling the Scheduler on WebSphere (page 107)
- Uninstalling the Scheduler on JBoss or Wildfly (page 108)
- Uninstalling the Scheduler on WebLogic (page 109)

**Uninstalling the Scheduler on WebSphere**

This section describes how to manually uninstall the Scheduler on WebSphere. Chapter 3 describes an alternate, automated way of doing so.

To uninstall the Scheduler on WebSphere:

1. Inside the Application Server, uninstall the "eSL Scheduler" application.
2. Delete the activation specification created when the Scheduler was installed.
3. Delete the JMS queue SchedulerRequestQueue.
4. If you used WebSphere SIB default messaging, remove the bus destination SchedulerQueueDestination.
5. If you used WebSphere MQ messaging, on the MQ server delete the queue APPROVEIT.SCHEDULER.QUEUE.
6. Delete the Data Source for the e-Mail Notification Manager™ Database.
7. Delete the JDBC Provider for the e-Mail Notification Manager™ Database (provided it’s not being used by any other application).
8. If you copied any JDBC driver files for the e-Mail Notification Manager™ Database when the Scheduler was installed, delete them (provided they’re not being used by any other application).
9. Delete the directory in which JMS queue messages were to be stored.
10. Delete the JVM system property that was specified when the Scheduler was installed.
11. Delete the contents of the e-Mail Notification Manager™ Database (but not the database itself).
Uninstalling the Scheduler on JBoss or Wildfly

12. If the e-Mail Notification Manager™ Database is not part of the Core Database, perform the following related steps:
   a. Make a backup copy of the e-Mail Notification Manager™ Database.

   ! If you do not have a backup copy of this database, you will permanently lose all its data. However, don't delete this database if it is part of the Core Database.

   b. Delete the e-Mail Notification Manager™ Database and all its contents.

13. Restart the Application Server.

   This procedure ends here.

Uninstalling the Scheduler on JBoss or Wildfly

This section describes how to manually uninstall the Scheduler on JBoss or Wildfly. Chapter 3 describes an alternate, automated way of doing so.

To uninstall the Scheduler on JBoss or Wildfly:

1. Inside the Application Server, delete the file "eSL-Scheduler.ear" from the deployment directory.

2. Delete the JMS queue SchedulerRequestQueue.

3. Delete the Data Source for the e-Mail Notification Manager™ Database.

4. Delete the JDBC Provider for the e-Mail Notification Manager™ Database (provided it’s not being used by any other application).

5. If you copied any JDBC driver files for the e-Mail Notification Manager™ Database when the Scheduler was installed, delete them (provided they’re not being used by any other application).

6. Delete the directory in which JMS queue messages were to be stored.

7. Delete the JVM system property that was specified when the Scheduler was installed.

8. Delete the contents of the e-Mail Notification Manager™ Database (but not the database itself).

9. If the e-Mail Notification Manager™ Database is not part of the Core Database, perform the following related steps:
   a. Make a backup copy of the e-Mail Notification Manager™ Database.

   ! If you do not have a backup copy of this database, you will permanently lose all its data. However, don't delete this database if it is part of the Core Database.
b. Delete the e-Mail Notification Manager™ Database and all its contents.

10. Restart the Application Server.

This procedure ends here.

Uninstalling the Scheduler on WebLogic

To uninstall the Scheduler on WebLogic:

1. Inside the Application Server, delete the application whose default name was "eSL Scheduler".

2. Delete the JMS queue SchedulerRequestQueue.

3. Delete the Data Source for the e-Mail Notification Manager™ Database.

4. Delete the JDBC Provider for the e-Mail Notification Manager™ Database (provided it’s not being used by any other application).

5. If you were not using an Oracle database when the Scheduler was installed, and if you copied any JDBC driver files for the e-Mail Notification Manager™ Database at that time, delete them (provided they’re not being used by any other application).

6. Delete the directory in which JMS queue messages were to be stored.

7. Delete the JVM system property that was specified when the Scheduler was installed.

8. Delete the contents of the e-Mail Notification Manager™ Database (but not the database itself).

9. If the e-Mail Notification Manager™ Database is not part of the Core Database, perform the following related steps:

   a. Make a backup copy of the e-Mail Notification Manager™ Database.

   If you do not have a backup copy of this database, you will permanently lose all its data. However, don’t delete this database if it is part of the Core Database.

   b. Delete the e-Mail Notification Manager™ Database and all its contents.

10. Restart the Application Server.

    This procedure ends here.

Uninstalling the Notifier

To uninstall the Notifier, you need perform only one of the following procedures (depending on your Application Server):

- Uninstalling the Notifier on WebSphere (page 110)
- Uninstalling the Notifier on JBoss or Wildfly (page 110)
To uninstall the Notifier on WebSphere:

1. Inside the Application Server, uninstall the "eSL Notifier" application.
2. Delete the activation specification created during the installation of the Notifier.
3. Delete the JMS queue `NotifierRequestQueue`.
4. If you used WebSphere SIB default messaging, remove the bus destination `NotificationQueueDestination`.
5. If you used WebSphere MQ messaging, on the MQ server delete the queue `APPROVEIT.NOTIFIER.REQUEST.QUEUE`.
6. Delete the directory in which JMS queue messages were to be stored.
7. Delete the JVM system property that was specified when the Notifier was installed.
8. Restart the Application Server.

This procedure ends here.

To uninstall the Notifier on JBoss or Wildfly:

1. Inside the Application Server, delete the file "eSL-Notifier.ear" from the deployment directory.
2. Delete the JMS queue `NotifierRequestQueue`.
3. Delete the directory in which JMS queue messages were to be stored.
4. Delete the JVM system property that was specified when the Notifier was installed.
5. Restart the Application Server.

This procedure ends here.
name was "eSL Notifier".
2. Delete the JMS queue NotifierRequestQueue.
3. Delete the directory in which JMS queue messages were to be stored.
4. Delete the JVM system property that was specified when the Notifier was installed.
5. Restart the Application Server.

This procedure ends here.
CHAPTER 11: Deploying PDF Document Engines

The PDF Document Engines perform all operations that manipulate PDF documents (signing, etc.). This chapter describes:

• Using Multiple PDF Document Engines (page 112)
• Tenant and Non-Tenant Environments (page 112)
• Installing PDF Document Engines (page 113)
• Uninstalling PDF Document Engines (page 157)

This chapter's procedures assume that your environment meets all relevant requirements in Chapter 2.

Using Multiple PDF Document Engines

OneSpan Sign can be deployed with multiple PDF Document Engines in a clustered environment. How many you should use depends on the load the OneSpan Sign system will have to process. This in turn depends on factors such as:

• The average number of users per hour, as well as the peak load
• The number of documents
• The size of documents
• The number of signers per document
• The number of signatures per document

Tenant and Non-Tenant Environments

Chapter 2 introduced the concept of tenants; see Tenants (page 24).

PDF Document Engines can work with one or more tenants (a "tenant environment") and/or with no tenants (the "non-tenant environment" that characterized OneSpan Sign before version 5.1 – i.e., the "backward-compatibility" option).

PDF Document Engines are configured through the parameters in Table 11-2.

If you configure Table 11-2's parameters for a tenant environment:

• If a parameter is not in the tenant's parameter list, that parameter's default value for the tenant will be taken from the value defined by the System tenant, provided the System tenant defines that parameter's value.
• If a parameter is not in the tenant's parameter list or in the System tenant's parameter list, its default value will be empty.
• If a parameter is in the tenant's parameter list but its value is empty, that parameter's value for the tenant will remain empty.
If you are in a tenant environment, you must specify all the parameters in Table 11-2 exclusively via the Admin Console (see Managing Settings in Chapter 3 of the Platform Administrator’s Guide). For more on tenant environments, see Appendix G.

If you are in a non-tenant environment, you must specify all the parameters in Table 11-2 exclusively via the file "DocumentEngineSettings.props".

For Linux, that file is in the directory `<DE deployment directory>/silanis.
For Windows, that file is in the directory `<DE deployment directory>/services/DEService.

Installing PDF Document Engines

To sign and tamperseal PDF documents, and to perform necessary cryptographic functions, the PDF Document Engine needs a digital certificate with an associated private key (OneSpan Sign tamperseals documents so it can detect any subsequent modifications). The certificate and key are packaged in multiple PEM files (PEM = Privacy Enhanced Mail). Starting with OneSpan Sign 5.1, deploying the PDF Document Engine with a CPS file (file extension = cps) is no longer supported. You must instead use PEM files, which began to be supported in OneSpan Sign 4.0.

Chapter 5 of the Platform Administrator’s Guide describes how to create PEM files. You can create a PEM file using either a self-signed certificate or a third-party certificate.

**VERY IMPORTANT**

You might use a self-signed certificate in a PEM file to perform development and testing activities. However, using self-signed certificates must be strictly limited to development and testing purposes only. For production deployment, you must use a managed certificate issued by a third-party provider such as VeriSign. You must not use self-signed certificates in a production OneSpan Sign system. The use of managed certificates enables you to be notified by the third-party provider about the status of your digital certificates, including expiry and revocation.

To ensure that your certificate can be validated by Adobe Reader, we strongly recommend using an AATL (Adobe Approved Trust List) certificate.

OneSpan is not responsible for the digital certificates you use with OneSpan Sign, whether self-signed or provided by a third-party.
To install a PDF Document Engine, sequentially perform the following procedures:

1. One of the following:
   • Installing a New PDF Document Engine on Linux (page 114)
   • Installing a New Doc Engine on Windows with Apache (page 123)
   • Installing a New Doc Engine on Windows with IIS (page 126)
   • Upgrading a PDF Document Engine on Linux (page 130)
   • Upgrading a PDF Document Engine on Windows (page 130)

2. Configuring a PDF Document Engine (page 131)

⚠️ Until you have performed the procedure Configuring the PDF Document Engine, the PDF Document Engine will not be ready to work in your production environment.

If you want to deploy more than one PDF Document Engine, repeat the above procedures on the machine of each additional PDF Document Engine.

**Installing a New PDF Document Engine on Linux**

**Prerequisites**

- The procedure Installing the Admin Console (page 65) has been performed.
- The following two users are available to perform the installation: (1) the application user; (2) the superuser (username = root).
- The application user has READ and WRITE privileges on the directory specified by the parameter CachePath, which appears in Table 11-2.
- The application user has READ and WRITE privileges to the PDF Document Engine’s "deployment directory" — i.e., the directory to which the PDF Document Engine will be deployed.
- The application user has the necessary permissions to access local resources.
- Apache HTTP Server is installed on the machine where you want to install the PDF Document Engine.
- Apache HTTP Server is configured with the following Apache settings:
  • KeepAlive = Off
  • MaxRequestsPerChild = 10000
  • Mode = prefork. You can determine your mode via a command like httpd -V.
- You have opened the appropriate port in your firewall for use by the new PDF Document Engine (e.g., port 80).
- The "SELinux" policy is set to Permissive, Disabled, or Enforcing. Enforcing mode provides enhanced security. If you are using that mode, the superuser must do the following:
  a. In the file /etc/selinux/config, ensure that SELINUX=enforcing.
b. Reboot the server.

c. From the "install" folder, run the following command:

```
./de_selinux.sh <DE deployment directory> <cache>
```

Here `<DE deployment directory>` is the full path of the PDF Document Engine deployment directory, and `<cache>` is the directory specified by the parameter `CachePath` in Table 11-2.

- You have identified the location of the PDF Document Engine tar file "install.tar.gz" within the software provided by OneSpan.
- To avoid missing or inappropriately arranged text, we recommend that you install on the PDF Document Engine server: (1) the standard core set of Linus fonts; (2) the Microsoft Office core font package; (3) any additional fonts you may need. Fonts are usually installed in the directory `/usr/share/fonts`.
- If you plan to use the PDF Document Engine in a tenant environment, you have configured the PDF Document Engine parameters in Table C-20 of the Platform Administrator's Guide (see Managing Settings in Chapter 3 of that guide).
- The following Operating System libraries and packages have been included on your machine:
  - `libfreetype.so.6`
  - `libz.so.1`
  - `libssl3.so.0`
  - `libjpeg.so.62`
  - `libpng12.so.0`
  - `libfontconfig.so.1`
  - `libXext.so.6`
  - `libXt.so.6`
  - `libbz2.so.1`
  - `libxml2.so.2`
  - `libSM.so.6`
  - `libICE.so.6`
  - `libX11.so.6`
  - `libXrender.so.1`
  - `libgomp.so.1`
  - `libidn.so.11`
  - `libopenjp2.so.7`
  - `libpangoft2-1.0.so.0`
  - `libpangocairo-1.0.so.0`
- If you are using RHEL/CentOS 6, the following Operating System libraries and packages have been included on your machine:
  - `pango-1.28.1-11.el6.x86_64`
  - `gstreamer-0.10.29-1.el6.x86_64`
  - `gstreamer-plugins-base-0.10.29-2.el6.x86_64`
• **compat-expat1-1.95.8-8.el6.x86_64. Note:** (1) This package can be found in RedHat’s official portal; (2) this package contains the required library "libexpat.so.0".

• If you are using RHEL/CentOS 7, the following Operating System libraries and packages have been included on your machine:
  • graphite2-1.3.6-1.el7_2.x86_64
  • harfbuzz-1.3.2-1.el7.x86_64
  • cairo-1.14.2-1.el7.x86_64
  • pango-1.36.8-2.el7.x86_64
  • gstreamer-0.10.36-7.el7.x86_64
  • gstreamer-tools-0.10.36-7.el7.x86_64
  • gstreamer-plugins-base-0.10.36-7.el7.x86_64
  • compat-expat1-1.95.8-8.el6.x86_64. **Note:** (1) You must download this package from RedHat’s official portal; (2) extract from this package the required library "libexpat.so.0", and put it in the *silanis* sub-directory of the deployment directory for the PDF Document Engine.

• If you want to use the *Electronic Evidence™ Export Utility*, the superuser has ensured that the Operating System will expose a display named 99. We recommend that the superuser do so as follows:
  a. By doing the following, install the "X virtual frame buffer" called **Xvfb** on the machine where the PDF Document Engine will be deployed:
     • If you are using RHEL/CentOS 6, run the following commands:
       
       subscription-manager repos --enable rhel-6-server-optional-rpms
       
       yum install xorg-x11-server-Xvfb
     
     • If you are using RHEL/CentOS 7, run the following commands:
       
       subscription-manager repos --enable rhel-7-server-optional-rpms
       
       yum install xorg-x11-server-Xvfb
   
   b. In the directory "/etc", create a file called "X99.hosts" that contains just the following line:
      
      <Apache_username>
      
      This is the username of the application user (e.g., silanis).
   
   c. Open the file "rc.local" in the directory "/etc/rc.d".
d. Add the following line to the end of that file:

```
Xvfb :99 -auth /etc/X99.hosts -screen 0 1x1x16 &
```

The preceding line instructs `Xvfb` to: (1) create a new virtual frame buffer; (2) publish that buffer as display 99; (3) attach 1 virtual screen to the display; (4) give that screen a resolution of 1 pixel by 1 pixel, and a color depth of 16 bits per pixel; (5) authorize the Apache user to use the frame-buffer driver. Note that the physical size of the display (in centimeters, say) is not important because the rendering is done off-screen.

e. Save and close the file.

f. If you are using RHEL/CentOS 7, run the following commands:

```
chmod u+x /etc/rc.d/rc.local
systemctl start rc-local
```

g. Restart the machine.

**Action**

**To install a new PDF Document Engine on Linux:**

1. As the application user, log on to the machine where you want to install the PDF Document Engine.

2. Copy the PDF Document Engine tar file "install.tar.gz" to a suitable local directory.

3. Uncompress the tar file by executing the following command:

```
tar xvf install.tar.gz.
```

This creates a directory called "install".

4. Outside the "install" directory, create a directory where the deployment will take place. Use a command like "mkdir <DE deployment directory>", where `<DE deployment directory>` refers to the full path of the PDF Document Engine deployment directory.

5. From within the "install" directory, run the following command:

```
./de_install.sh <DE deployment directory>
```

If you installed Apache in a customized location, the preceding command requires a second argument that specifies the full pathname of Apache's top installation folder. If that argument is not specified, the system uses the argument's default value (`/etc/httpd`). The script "silanis.sh" will eventually copy the configuration file "silanis.conf" from the directory "<DE deployment directory>/utils" to the directory "<second argument>/conf.d". The latter location is where Apache will look for "silanis.conf".
If you wish, you can activate the "silent configuration" option to ensure that the software asks you no further questions once you enter the above command. To do so, add the final argument -y at the end of the command. The script will then run without interruption, regardless of whether you included the Apache arguments from the above warning.

6. Read the license agreement "EULA.txt", which is in the "install" directory. If you agree with it, type "yes" on the command line.

7. If you are using RHEL/CentOS 6, and if you are in a tenant environment, open the file <DE deployment directory>/utils/silanis.sh, find the following lines, and complete the specification of RepoAddress:

```
export RepoAddress=<Filesystem or URL of the Remote Properties Repository>
export RepoBranch=documentengine
export RepoPath=<DE deployment directory>/silanis/repo
```

The above default setting for RepoBranch is strongly recommended, but you have the ability to change it if you wish (see Table 11-1).

The above default setting for RepoPath can be changed if you wish (see Table 11-1).

8. If you are using RHEL/CentOS 7, and if you are in a tenant environment, open the file <DE deployment directory>/utils/DEMultitenancy.env, find the following lines, and complete the specification of RepoAddress:

```
RepoAddress=<Filesystem or URL of the Remote Properties Repository>
RepoBranch=documentengine
RepoPath=<DE deployment directory>/silanis/repo
```

The above default setting for RepoBranch is strongly recommended, but you have the ability to change it if you wish (see Table 11-1).

The above default setting for RepoPath can be changed if you wish (see Table 11-1).

9. If you want to deactivate FIPS mode (which is activated by default), do one of the following:

   a. If you are using RHEL/CentOS 6, open the file <DE deployment directory>/utils/silanis.sh, and add the following line:

```
export DE_FIPS_MODE=OFF
```
b. If you are using RHEL/CentOS 7, open the file <DE deployment directory>/utils/DEMultitenancy.env, and add the following line:

```
DE_FIPS_MODE=OFF
```

💡 For more on FIPS mode, see Appendix F.

10. Optional: For performance reasons, the Electronic Evidence™ Export Utility does not by default render PDF output of the JPEG images in source HTML files. If you want that functionality, the application user should run the following command:

```
./jpegpath.sh <DE deployment directory>
```

The script "jpegpath.sh" is in the "install" directory.

⚠️ Ensure that you keep a backup of the original "httpd.conf" file that came with your Apache installation.

11. The superuser should edit the Apache configuration file "httpd.conf" in the directory "<Apache installation folder>/conf" to ensure that:

a. ServerRoot specifies the Apache installation folder.

b. The Apache username and group name are appropriately specified .

c. If you are using Apache 2.2.x, the statement "Include conf.d/*conf" is present.

d. If you are using Apache 2.4.x, the statement "IncludeOptional conf.d/*conf" is present.

12. Optional: If you want your version of Apache that runs the PDF Document Engine to have no dynamic dependencies on the system’s OpenSSL libraries, do the following:

a. If a file named php.conf exists in the Apache directory <Apache installation directory>/conf.d, remove it from your Apache installation.

b. If Apache has a directory named <Apache installation directory>/conf.modules.d, and if that directory has a file named 10-php.conf, remove that file from your Apache installation.

c. Do not load the Apache module <Apache installation directory>/modules/mod_ssl.so from Apache's configuration file <Apache installation directory>/conf/httpd.conf. **Warning:** You must load that module from that file if your Apache
installation needs SSL communications that are independent of the PDF Document Engine.

The PDF Document Engine is built with a static link to its own FIPS-enabled version of OpenSSL.

13. If you are using RHEL/CentOS 6, and you want the PDF Document Engine and DEMultitenancy to be started on Apache by the superuser (as recommended), the superuser should do the following:
   a. Add the following line to the top of the start function inside the Apache initialization script (e.g., "/etc/init.d/httpd"):
      
      source <DE deployment directory>/utils/silanis.sh
   b. Add the following line to the top of the stop function inside the Apache initialization script:
      
      initctl stop DEMultitenancy
   c. Add the following line to the top of the reload function inside the Apache initialization script (e.g., "/etc/init.d/httpd"):
      
      source <DE deployment directory>/utils/dem.sh

14. If you are using RHEL/CentOS 6 and you want the PDF Document Engine and DEMultitenancy to be started on Apache by the application user (not recommended):
   a. The superuser should run the script "admininstall6.sh". For details on how to do so, see Starting the Doc Engine on RHEL/CentOS 6 as an Application User (page 322).
   b. The application user should add the following line to the top of the start function inside the Apache initialization script (e.g., "/etc/init.d/httpd"):
      
      source <DE deployment directory>/utils/silanis.sh
   c. The application user should add the following line to the top of the stop function inside the Apache initialization script:
      
      initctl stop DEMultitenancy
d. The *application user* should add the following line to the top of the 
reload function inside the Apache initialization script (e.g., 
"/etc/init.d/httpd"):

```
source <DE deployment directory>/utils/dem.sh
```

15. **Optional:** If you want to start Apache now, if Apache will be started by 
system services, and if you are using RHEL/CentOS 6, run the 
command "service httpd start".

⚠️ **If you don’t want to start Apache now, you must do so 
before running the PDF Document Engine.**

💡 **If you ever want to stop Apache, run the command "service 
httpd stop".**

⚠️ **If you want the PDF Document Engine to be started on 
Apache by the *application user* (not recommended), do 
**not** perform the next step.**

16. If you are using RHEL/CentOS 7, and you want the PDF Document 
Engine to be started on Apache by the *superuser* (as recommended), 
the superuser should:

a. **Copy the files** <DE deployment directory>/utils/ 
DEHttpd.service and <DE deployment directory>/utils/ 
DEMultitenancy.service **to the services directory of the 
Operating System** (e.g., /usr/lib/systemd/system).

b. To ensure that all service configurations are up-to-date, run the 
command "systemctl daemon-reload".

c. **Optional:** If you want to start Apache now, the superuser should 
run the command "systemctl start DEHttpd".

⚠️ **If you don’t want to start Apache now, you must 
do so before running the PDF Document Engine.**

💡 **If you ever want to stop Apache, run the command 
"systemctl stop DEHttpd".**

17. Verify your newly configured installation by using a browser to go to 
the following URL:

- http(s)://<server>:<port>/axis2/services
Here `<server>` denotes the server where the PDF Document Engine is installed, and `<port>` denotes the port used by Apache. You should see a list of deployed services, including `DEService`.

You must configure the Core Component to communicate with the PDF Document Engine using the Platform Administrator’s Guide (see Table C-20 in that guide). If you don’t do this, the PDF Document Engine will not be able to process PDFs.

If this procedure was successful, the PDF Document Engine can now be started on Apache by the `superuser`. We do not recommend allowing the `application user` to start it, since since Apache will become less secure. If you nonetheless want to do so, see Appendix J.

This procedure ends here.

### Table 11-1: Environment variables for the PDF Document Engine

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>RepoAddress</td>
<td>URL or location of the Remote Properties Repository&lt;br&gt;Note: If this parameter is specified, the PDF Document Engine is considered to be configured for a &quot;tenant environment&quot;. If this parameter is empty, the PDF Document Engine is considered to be configured for a &quot;non-tenant environment&quot;.</td>
<td>The URL or location can be expressed using the local File System or one of three transport protocols (HTTP, HTTPS, or SSH). Example: https://&lt;server&gt;:&lt;port&gt;/resources-master/storage</td>
</tr>
<tr>
<td>RepoBranch</td>
<td>Branch of the Remote Properties Repository where PDF Document Engine properties are stored</td>
<td>Default: <code>documentengine</code></td>
</tr>
<tr>
<td>RepoPath</td>
<td>Absolute path to the top-level directory in which PDF Document Engine properties are stored for all tenants</td>
<td>Default for Linux: <code>&lt;DE deployment directory&gt;/silanis/repo</code>&lt;br&gt;Default for Windows: <code>&lt;DE deployment directory&gt;/services/DEService/repo</code></td>
</tr>
</tbody>
</table>
Installing a New Doc Engine on Windows with Apache

Prerequisites

- You plan to install a new PDF Document Engine on Windows using Apache HTTP Server.
- The procedure Installing the Admin Console (page 65) has been performed.
- The user who will run the PDF Document Engine service is part of the Administrator’s group.
- The user who will run the PDF Document Engine service has the MODIFY permission on the directory specified by the parameter CachePath in Table 11-2.
- The PDF Document Engine installation directory has been copied from the software provided by OneSpan to the machine where you want to install the PDF Document Engine.
- You have identified the location of the PDF Document Engine setup program.
- You have opened the appropriate port in your firewall for use by the new PDF Document Engine (e.g., port 80).

If an installation of IIS that could potentially run another application is already on your PDF Document Engine server, ensure that: (1) Apache and IIS never share the same port; (2) you never simultaneously run the same PDF Document Engine on Apache and IIS, even if they’re on different ports.

- If you plan to use the PDF Document Engine in a tenant environment, you have configured the PDF Document Engine parameters in Table C-20 of the Platform Administrator’s Guide (see Managing Settings in Chapter 3 of that guide).
- If you are using Microsoft Windows Server 2012, you have installed .NET Framework 3.5 as follows:
  a. Open the Windows Server Manager.
  b. Select Add Roles and Features. A Roles Wizard starts.
  c. At the Installation Type step, select Role based installation.
  d. At the Local Server step, select your local PDF Document Engine server.
  e. At the Features step, find the WebServer section.
  g. Restart your machine.
- If you are using Microsoft Windows Server 2012 R2, you have installed .NET Framework 3.5 as follows:
  a. You have a Windows-8-family installation medium for the 64-bit version of Microsoft Windows Server 2012 R2. That medium could be a DVD drive, or even a mounted ISO image from a Windows-8-family installation.
b. An Administrator should run the following command:

```
dism.exe /online /enable-feature /featurename:NetFX3/
All/Source:C:\sources\sxs /LimitAccess
```

Here `C:\sources\sxs` is the SxS path of your Windows-8-family installation medium.

c. Restart your machine.

If your server has Windows Update access, neither an installation medium nor the executable `dism.exe` is required to install .Net Framework 3.5. For alternative instructions in that case, see https://technet.microsoft.com/en-ca/library/dn482071.aspx.

• To avoid missing or inappropriately arranged text, we recommend that you install on the PDF Document Engine server: (1) the Microsoft Office core font package; (2) any additional fonts you may need.

**Action**

**To install a new PDF Document Engine on Windows using Apache HTTP Server:**

1. Log on to the machine where you want to install the PDF Document Engine.
2. Run the relevant setup program called "setup.exe". The installer’s Welcome screen appears.
3. Click Next. The License Agreement dialog box appears.
4. Read the license agreement. Select I accept the license agreement, and click Next. The Platform Selection dialog box appears.
5. Select Install on Apache Server.
6. Do one of the following:
   • If you plan to use the PDF Document Engine in a non-tenant environment, select Local Document Engine Properties.
   • If you plan to use the PDF Document Engine in a tenant environment:

   We do not recommend using a repository branch other than the default one (documentengine). If you want to do so, on the server where you are installing the PDF Document Engine you must define an Apache environment variable called RepoBranch. For instructions on setting environment variables, see your Operating System’s documentation.

   There is also an environment variable called RepoPath (see Table 11-1). It can be changed in the same manner as RepoBranch.
B. In the provided box, type the **URL of the Remote Properties Repository**. This step assumes that you have set up the *Remote Properties Repository*, and that the PDF Document Engine’s server can access it via its URL.

If a `RepoAddress` environment variable already exists, the Installer will automatically populate this box with the value specified for that variable. Thus in this case, a suggested URL appears in this box. You may accept that suggestion, or type another URL. In either case, the PDF Document Engine Uninstaller will never delete the `RepoAddress` environment variable.

7. Click **Next**. The *Destination Folder* dialog box appears.

8. Optional: If you want to install the PDF Document Engine in a directory different from the default, click **Browse** and select a different directory.

9. If you are ready to install, click **Next**.

10. A new window states that you are about to trigger the *Microsoft Visual C++ 2008 Redistributable Installer*. Click **Next**.

11. When that installer has finished, the Apache Installer automatically starts. Follow its instructions until it finishes.

12. The *Updating System* window appears. When its progress indicator is full, the *Successfully Installed* dialog box appears. Click **Finish**.

13. You will be prompted to restart the system. Click **Yes**. All affected system files are updated. This will also enable the Windows Service called *Silanis e-SE Remote Document Engine Properties* to start automatically after you start your Apache HTTP Server.

14. If you don’t reboot your machine now, you must manually start this Windows Service from the *Windows Service Manager* before you start your Apache HTTP Server.

15. If you are running Apache HTTP Server as a service, change the user who will run this service to the user who will run the PDF Document Engine service.

16. Ensure that the user who will run the PDF Document Engine service has the MODIFY permission on the Apache logs directory.

17. If you want to deactivate FIPS mode (which is activated by default):
   a. Set the following system environment variable:

   ```plaintext
   DE_FIPS_MODE=OFF
   ```

   To do so, click **Start**, right-click **Computer**, select **Properties**, click **Advanced system settings > Environment Variables**, under
Installing a New Doc Engine on Windows with IIS

System variables click New, type DE_FIPS_MODE as the Variable name and OFF as the Variable value, and finally click OK > OK.

b. If the hosting Apache HTTP server is running, restart it.

For more on FIPS mode, see Appendix F.

18. Optional: For performance reasons, the Electronic Evidence™ Export Utility does not by default render PDF output of the JPEG images in source HTML files. If you want that functionality:

a. Go to the PDF Document Engine’s deployment directory.

b. Go to the subdirectory
"/services/DEservice/plugins/imageformats".

c. Remove the ".bak" suffix from any "jpeg*" DLL file in that directory.

You must configure the Core Component to communicate with the PDF Document Engine using the Platform Administrator’s Guide (see Table C-20 in that guide). If you don’t do this, the PDF Document Engine will not be able to process PDFs.

This procedure ends here.

Installing a New Doc Engine on Windows with IIS

Using Apache HTTP Server is preferable to using IIS HTTP Server. However, if you plan to use IIS HTTP Server, perform the procedure in this section.

Prerequisites

• You plan to install a new PDF Document Engine on Windows using IIS HTTP Server.

• The procedure Installing the Admin Console (page 65) has been performed.

• The user who will run the PDF Document Engine service is part of the Administrator’s group.

• The user who will run the PDF Document Engine service has the MODIFY permission on the directory specified by the parameter CachePath in Table 11-2.

• The PDF Document Engine installation directory has been copied from the software provided by OneSpan to the machine where you want to install the PDF Document Engine.

• You have identified the location of the PDF Document Engine setup program.

• You have opened the appropriate port in your firewall for use by the
new PDF Document Engine (e.g., port 80).

If an installation of Apache that could potentially run another application is already on your PDF Document Engine server, ensure that: (1) IIS and Apache never share the same port; (2) you never simultaneously run the same PDF Document Engine on IIS and Apache, even if they're on different ports.

- If you plan to use the PDF Document Engine in a tenant environment, you have configured the PDF Document Engine parameters in Table C-20 of the Platform Administrator’s Guide (see Managing Settings in Chapter 3 of that guide).
- If you are using Microsoft Windows Server 2012, you have installed .NET Framework 3.5 as follows:
  a. Open the Windows Server Manager.
  b. Select Add Roles and Features. A Roles Wizard starts.
  c. At the Installation Type step, select Role based installation.
  d. At the Local Server step, select your local PDF Document Engine server.
  e. At the Features step, find the WebServer section.
  g. Restart your machine.
- If you are using Microsoft Windows Server 2012 R2, you have installed .NET Framework 3.5 as follows:
  a. You have a Windows-8-family installation medium for the 64-bit version of Microsoft Windows Server 2012 R2. That medium could be a DVD drive, or even a mounted ISO image from a Windows-8-family installation.
  b. An Administrator should run the following command:

        dism.exe /online /enable-feature /featurename:NetFX3/All/Source:C:\sources\sxs /LimitAccess

     Here C:\sources\sxs is the SxS path of your Windows-8-family installation medium.
  c. Restart your machine.

     If your server has Windows Update access, neither an installation medium nor the executable dism.exe is required to install .Net Framework 3.5. For alternative instructions in that case, see https://technet.microsoft.com/en-ca/library/dn482071.aspx).

- IIS HTTP Server is installed on the machine where you want to install the PDF Document Engine.
- A dedicated IIS Web site with its own port has been created for the PDF Document Engine (not the IIS Default Web Site). The remainder of this chapter refers to that Web site as [IIS Website].

© 2020 - OneSpan North America Inc.     DG-OSS-72 - All rights reserved.
• **ISAPI Extensions** are allowed.

⚠️ Because **ISAPI Extensions** are configured as a wildcard, **axis** will process all calls to the **PDF Document Engine** port. We strongly recommend that you nonetheless dedicate this port **to PDF Document Engine SOAP calls only**. Other HTTP calls should not be sent to this port (e.g., calls for monitoring or security purposes). The one exception to this rule is the call "http(s):/ip-address/axis2/services", which can be used from a browser to verify that the **PDF Document Engine is running**.

• Perform the procedure in Appendix I.
• To avoid missing or inappropriately arranged text, we recommend that you install on the **PDF Document Engine server**: (1) the Microsoft Office core font package; (2) any additional fonts you may need.

**Action**

**To install a new PDF Document Engine on Windows using IIS HTTP Server:**

1. Log on to the machine where you want to install the **PDF Document Engine**.
2. Run the relevant setup program called "setup.exe". The installer’s **Welcome** screen appears.
3. Click **Next**. The **License Agreement** dialog box appears.
4. Read the license agreement. Select **I accept the license agreement**, and click **Next**. The **Platform Selection** dialog box appears.
5. Select **Install on Internet Information Services**.
6. Do one of the following:
   • If you plan to use the **PDF Document Engine** in a **non-tenant** environment, select **Local Document Engine Properties**.
   • If you plan to use the **PDF Document Engine** in a **tenant** environment:
     A. Select **Remote Document Engine Properties**.
        We do not recommend using a repository branch other than the default one (**documentengine**). If you want to do so, on the server where you are installing the **PDF Document Engine** you must define an IIS environment variable called **RepoBranch**. For instructions on setting environment variables, see your Operating System’s documentation.
     B. In the provided box, type the **URL of the Remote Properties Repository**. This step assumes that you have set up the
Installing a New Doc Engine on Windows with IIS

Remote Properties Repository, and that the PDF Document Engine’s server can access it via its URL.

If a RepoAddress environment variable already exists, the Installer will automatically populate this box with the value specified for that variable. Thus in this case, a suggested URL appears in this box. You may accept that suggestion, or type another URL. In either case, the PDF Document Engine Uninstaller will never delete the RepoAddress environment variable.

7. Click Next. The Destination Folder dialog box appears.
8. Optional: If you want to install the PDF Document Engine in a directory different from the default, click Browse and select a different directory.
9. If you are ready to install, click Next.
10. The Updating System window appears. When its progress indicator is full, the Successfully Installed dialog box appears. Click Finish.

The next step is highly recommended for IIS.

11. You will be prompted to restart the system. Click Yes. All affected system files are updated. This will also enable the Windows Service called Silanis e-SE Remote Document Engine Properties to start automatically after you start your IIS HTTP Server.

If you don’t reboot your machine now, you must manually start this Windows Service from the Windows Service Manager before you start your IIS HTTP Server.

12. If you want to deactivate FIPS mode (which is activated by default):
   a. Set the following system environment variable:

   \texttt{DE\_FIPS\_MODE=OFF}

   To do so, click Start, right-click Computer, select Properties, click Advanced system settings > Environment Variables, under System variables click New, type DE\_FIPS\_MODE as the Variable name and OFF as the Variable value, and finally click OK > OK.

   b. If the hosting IIS HTTP server is running, restart it.

For more on FIPS mode, see Appendix F.
Optional: For performance reasons, the Electronic Evidence™ Export Utility does not by default render PDF output of the JPEG images in source HTML files. If you want that functionality:

a. Go to the PDF Document Engine’s deployment directory.

b. Go to the subdirectory "/services/DEservice/plugins/imageformats".

c. Remove the ".bak" suffix from any "jpeg*" DLL file in that directory.

You must configure the Core Component to communicate with the PDF Document Engine using the Platform Administrator’s Guide (see Table C-20 in that guide). If you don’t do this, the PDF Document Engine will not be able to process PDFs.

This procedure ends here.

Upgrading a PDF Document Engine on Linux

To upgrade an existing PDF Document Engine on Linux:

1. Log on to the machine where the PDF Document Engine is installed.

2. Go to the directory where the PDF Document Engine is installed.

3. Backup that directory’s PEM files or CPS file to a safe location.

4. If Apache HTTP Server is running, stop it.

5. Uninstall the existing PDF Document Engine, using the procedure Uninstalling PDF Document Engines (page 157).

6. Manually restore the Apache configuration file "httpd.conf" to the state it was in before the PDF Document Engine was installed.


Some time after you perform Step 7, you may choose to update your Apache installation using the yum facility. If so, you may need to re-perform Step 11 to Step 17 of Installing a New PDF Document Engine on Linux.

To ensure that the upgraded PDF Document Engine works as expected, verify that it is still configured appropriately (see Table C-20 of the Platform Administrator’s Guide).

This procedure ends here.

Upgrading a PDF Document Engine on Windows

To upgrade an existing PDF Document Engine on Windows:

1. Log on to the machine where the PDF Document Engine is installed.

2. Go to the directory where the PDF Document Engine is installed.

3. Backup that directory’s PEM files or CPS file to a safe location.
4. If Apache or IIS HTTP Server is running, stop it.
5. Uninstall the existing PDF Document Engine, using the procedure Uninstalling PDF Document Engines (page 157).
6. If you are using Apache HTTP Server, perform the procedure Installing a New Doc Engine on Windows with Apache (page 123).
7. If you are using IIS HTTP Server, perform the procedure Installing a New Doc Engine on Windows with IIS (page 126).

To ensure that the upgraded PDF Document Engine works as expected, verify that it is still configured appropriately (see Table C-20 of the Platform Administrator’s Guide).

This procedure ends here.

Configuring a PDF Document Engine

Prerequisites

- The PDF Document Engine is not currently running.
- On the machine that will run the PDF Document Engine, one or more PEM files has been created (see Chapter 5 of the Platform Administrator’s Guide). A PEM file is required by the parameter CPSFile in Table 11-2.

A PEM (Privacy Enhanced Mail) file is a file with the extension pem that contains a private key or one or more certificates used during the signing or tampersealing of PDF documents.

To ensure that all PDF Document Engines use the same certificate to process documents, use the same "CPSFile" parameter for all engines.

- If you are using Linux, and want to use the tenant feature with the SSH protocol (see Table 11-1), the SSH private and public keys to your target server must be stored in the files /root/.ssh/id_rsa and /root/.ssh/id_rsa.pub, respectively. This assumes that the application user has been authorized to log into the SSH server using those keys.
- If you are using Linux, and you intend to use the "tenant" feature with the HTTPS protocol (see Table 11-1), the following certificates must be stored in the file /usr/local/ssl/cert.pem (in PEM format):
  - The certificate of the Web Server on which the Remote Properties Repository resides (see RepoAddress in Table 11-1)
  - Any associated intermediate certificates
  - The associated public-key root certificate issued by a Certificate Authority
- If you are using Windows, and want to use the "tenant" feature with the SSH protocol (see Table 11-1), the SSH private and public keys to your target server must be: (1) stored in the files id_rsa and id_rsa.pub, respectively; (2) placed in the directory where the PDF Document
Engine is installed. This assumes that the IIS/Apache user has been authorized to log into the SSH server using those keys.

- If you are using Windows, and want to use the "tenant" feature with the HTTPS protocol (see Table 11-1), the following certificates must be stored in the Certificate Store of the server that hosts the PDF Document Engine:
  - The certificate of the Web Server on which the Remote Properties Repository resides (see RepoAddress in Table 11-1)
  - Any associated intermediate certificates
  - The associated public-key root certificate issued by a Certificate Authority

You can sign documents using any combination of signing method (Click-to-Sign, Input Device, Certificate), Operating System (Linux, Windows), and signature type (Silanis, Adobe). However, only Adobe Signatures support timestamps.

The PDF Document Engine signs documents using the SHA-256RSA algorithm.

**Action**

If you are in a non-tenant environment, you must specify all the parameters in Table 11-2 exclusively via the file "DocumentEngineSettings.props". If you are in a tenant environment, you must specify all the parameters in Table 11-2 exclusively via the Admin Console (see Managing Settings in Chapter 3 of the Platform Administrator's Guide).

The following procedure asks you to specify various parameters in Table 11-2. Ensure that your specifications do not terminate with a white space.

If you are in a non-tenant environment and you change the file "DocumentEngineSettings.props" at any time (see Table 11-2), you must subsequently start or restart the Apache server to enable your changes to take effect. If the Apache server is already running, you must stop and restart it. When you change this file, its permission settings may be changed. To ensure that the Apache user can access the file, update those settings with the `chmod` command as root user to 755. For example: `chmod 755 DocumentEngineSettings.props`.

If the PDF Document Engine uses Apache HTTP Server, the Apache service may stop working if the credentials of the account under which it runs are changed. To prevent this, either configure that account to never expire, or consult your System Administrator.

**To configure a PDF Document Engine:**

1. Log on to the machine where the PDF Document Engine is installed.
2. If you are using IIS HTTP Server:
   a. Open the IIS configuration file "applicationHost.config" in the directory "%systemroot%/System32/inetsrv/config".
b. Change the tag `<serverRuntime />` to `<serverRuntime uploadReadAheadSize="52428800" />`.

c. Save and close the configuration file.

d. Let `[IIS User]` designate the user account that will run IIS. To provide IIS authentication for the `[IIS User]`:

   A. Ensure that `[IIS User]` is an account on the IIS HTTP Server, and thus has credentials recognized by the Operating System.

   B. Click **Start** > **Administrative Tools** > **Internet Information Services (IIS) Manager**. The corresponding window appears.

   C. In the main pane, click the **Authentication** icon. A new window appears.

   D. Right-click **Anonymous Authentication**, and select **Edit**. A new window appears.

   E. Select **Specific user**. A new window appears.

   F. Select **Set**, and type the existing credentials of the `[IIS User]`.

   G. Click **OK** (the associated window closes), and click **OK** again (the associated window closes).

   H. Return to the window you accessed in Step B.

e. Configure the `[IIS Website]` by doing the following:

   A. Expanding the tree in the left pane (**Connections**), under **Sites** click `[IIS Website]`.

   B. In the new right pane, right-click **Handler Mappings** > **Open Feature**.

   C. In the rightmost pane (**Actions**), click **Add Wildcard Script Map**. The corresponding dialog box appears.

   D. Using the browse button (...) beside the **Executable** field, select the DLL library "mod_axis2_IIS.dll" that is in the sub-directory "lib" of the directory where the PDF Document Engine service was installed.

   E. Click **Open**. The dialog box closes.

   F. In the **Name** field, type a suitable name (e.g., `axis`).

   G. Click **OK**. A new window appears.

   H. Click **Yes** to allow ISAPI extensions.

   I. In the main pane of the `[IIS Website]`, click the **Authentication** icon. A new window appears.

   J. Right-click **Anonymous Authentication**, and select **Edit**. A new window appears.

   K. Select **Specific user**. A new window appears.

   L. Select **Set**, and type credentials for the user account. They must be the same credentials as in Step 2d.
M. Click OK (the associated window closes), and click OK again (the associated window closes).

f. If you are using IIS 8.0 or 8.5, continue configuring the [IIS Website] by doing the following:

A. In the Handler Mappings section, click the handler you just mapped.
B. Select Edit on the right pane. A new window appears.
C. Click Request Restrictions. A new window appears.
D. Clear the checked radio button that restricts the handler to being a file.
E. Click OK. An ISAPI settings window appears.
F. Select Yes.
G. In the Handler Mappings section, click the handler just mapped.
H. Select Edit Feature Permissions on the right pane. A new window appears.
I. Ensure that all 3 boxes are selected (e.g., Read, Write, Execute) and click OK.

After you've cleared this button, you will not be able to return to this window until you've removed the handler and created a new one.

E. Click OK. An ISAPI settings window appears.

F. Select Yes.

G. In the Handler Mappings section, click the handler just mapped.

H. Select Edit Feature Permissions on the right pane. A new window appears.

I. Ensure that all 3 boxes are selected (e.g., Read, Write, Execute) and click OK.

To reliably restart IIS, do the following:

A. Ensure that the [IIS Website], your Application Pool, and the IIS Manager are stopped. In each case, the Stop command is in the right pane.
B. Return to the main IIS Manager window.
C. In the right pane, click Start. The IIS Manager restarts.
D. In the left pane, click Application Pools, and select your Application Pool (e.g., DefaultAppPool).
E. In the right pane, click Start. Your Application Pool restarts.
F. In the left pane, click [IIS Website].
G. In the right pane, click Start. The [IIS Website] restarts.

3. Specify the parameter "CPSFile" in Table 11-2.
4. If you are in a tenant environment, use the Admin Console to change the value of the parameter "CPSPassword" in Table 11-2 (see the Platform Administrator's Guide).
5. If you are in a non-tenant environment, do the following to change the password of the file specified by the parameter "CPSFile" in Table 11-2:

   a. Open a command window, and navigate to the relevant directory. For Linux, that directory is `<DE deployment directory>/silanis`. For Windows, that directory is `<DE deployment directory>/services/DEService`.

   b. If you are using Linux:
      
      A. Ensure that the "ChangePassword" file is executable by running the following command:
      ```
      chmod +x ChangePassword
      ```
      
      B. Ensure that the "ScriptPassword" file is executable by running the following command:
      ```
      chmod +x ScriptPassword
      ```
      
      C. Run the following command:
      ```
      ScriptPassword [CPSPassword] [RampartPassword]
      ```
      > When the command has executed, the following message appears below the command line: The PDF Document Engine’s properties file was successfully updated.

   c. If you are using Windows, run the following command:
      ```
      ChangePassword -s [CPSPassword] -r [RampartPassword]
      ```
      > Replace [CPSPassword] and [RampartPassword] with real values. For descriptions of those arguments, see Table 11-2.
      > When the command has executed, the following message appears below the command line:
      > The PDF Document Engine’s properties file was successfully updated.

   d. Close the command-line console.

6. Correctly specify the values of all desired parameters in Table 11-2 (in addition to "CPSFile" and "CPSPassword", which were specified in previous steps).

   If you are in a tenant environment, and if you want to see your tenant’s configuration settings in the file `decoro.log`, you must use the Admin Console to set the CoreTrace parameter for the System Tenant equal to DEBUG1.

7. Optional: If you want to change the logging behavior of the PDF Document Engine, see the section Logging for PDF Document Engines in Appendix B.
8. **Optional:** If you want to enable SOAP security for the PDF Document Engine’s SOAP messages:
   a. Locate in the directory "[<DE deployment directory>]//services/DEService" the file called "services.xml" that describes the PDF Document Engine’s Web service.
   b. Open that file, and uncomment the "rampart module" and "Policy" elements.
   c. Search for a line of the following form:
      ```xml
      <rampc:PasswordCallbackClass/></rampc:PasswordCallbackClass>
      ```
   d. If the PDF Document Engine is installed on Windows, make the line you found specify the location of the "DESecurity.dll" file by ensuring that it has the following form:
      ```xml
      <rampc:PasswordCallbackClass>
      <DE deployment directory>/services/DEService/DESecurity.dll
      </rampc:PasswordCallbackClass>
      ```
   e. If the PDF Document Engine is on Linux, make the line you found specify the location of the "libDESecurity.so" file by ensuring that it has the following form:
      ```xml
      <rampc:PasswordCallbackClass>
      <DE deployment directory>/services/DEService/libDESecurity.so
      </rampc:PasswordCallbackClass>
      ```
   f. Save and close the file.
   g. Locate in the directory "<DE deployment directory>/modules/rampart" the file called "module.xml".
   h. Open that file.
   i. Comment the "outflow" and "outfaultflow" XML tags.
   j. Save and close the file.

   For more on SOAP security, see this chapter’s section called *Enabling Soap Security*.

9. **Optional:** If you want to enhance network security between the PDF Document Engine and the Core Component, encrypt their communications channel.

10. When you start Apache or IIS HTTP Server, verify your newly configured installation by using a browser to go to the following URL:
    - http(s)://<server>[:<port>]/axis2/services

    Here `<server>` denotes the server where the PDF Document Engine is installed, and `<port>` denotes the port used by Apache or IIS HTTP Server. You should see a list of deployed services, including `DEService`.

    This procedure ends here.
<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPSFile</td>
<td>The full path of a *.pem file that contains the PDF Document Engine’s private key</td>
<td>If you’re in a non-tenant environment, the value must be a valid filesystem path. If you’re in a tenant environment, the value can be either a valid filesystem path (including the filename), or the filename alone. Because this file contains your private key, we strongly recommend that you keep it in a secure location by using the full filesystem path. If you use the filename alone, this file must be placed in the Remote Properties Repository for PDF Document Engine properties associated with each tenant (where it may be exposed to other users). That repository resides in the tenant’s Resources directory.</td>
</tr>
</tbody>
</table>

**Note:** (1) This file can be used only with Adobe Signatures, regardless of the Operating System; (2) the PDF Document Engine uses the OpenSSL Cryptographic Module.  

**Note:** If you have a Gemalto HSM deployment that uses the PKCS#11 feature of the PDF Document Engine, you must use the following format for this parameter:  

- `<Key Handle PEM file>#<Absolute Path>/libCryptoki2.so#<slot id>`  

However, if you are using slot id = 1 (the default value), and if the `libCryptoki` library is in its "Luna Client installed location", the format you should use for this parameter is simply `<Key Handle PEM file>`.
<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPSPassword</td>
<td>For a non-HSM deployment, this parameter is the encrypted password of the CPSFile described in the previous row.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For a Gemalto HSM 6 deployment that uses the PKCS#11 feature of the PDF Document Engine, this is the encrypted password of your HSM partition.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For a Gemalto HSM 7 deployment (Cloud or on-premises), this is the encrypted password of the Crypto Officer mentioned in Step 11 of Gemalto’s Cloud HSM-7 (DPOD). (page 238).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For an AWS Cloud HSM deployment, this is the encrypted password described in Step 23 of the procedure Amazon Web Services (AWS) (page 241).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This encrypted password is entered in Step 4 or Step 5 of the above procedure.</td>
<td></td>
</tr>
<tr>
<td>CachePath</td>
<td>Path of the root directory where the PDF Document Engine stores temporary files (incoming documents, outgoing documents or images, etc.)</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Warning</strong>: The user who will run the PDF Document Engine service must have READ and WRITE privileges on this directory.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The value should be a valid filesystem path.</td>
<td></td>
</tr>
<tr>
<td>CacheRetain</td>
<td>A flag that determines if a request’s temporary files will be retained after the request</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: This parameter is optional.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid values:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 1 (retain)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 0 (do not retain)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: 0</td>
<td></td>
</tr>
<tr>
<td>PARAMETER</td>
<td>DESCRIPTION</td>
<td>VALUES</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>RampartUsername</td>
<td>Username validated by the PDF Document Engine when it authenticates requests. If security is enabled, only requests providing the correct username and password will be served.</td>
<td>No default value</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: This parameter is optional unless SOAP security is enabled (see Step 8 above). For more on SOAP security, see this chapter’s section called <em>Enabling Soap Security</em>.</td>
<td></td>
</tr>
<tr>
<td>RampartPassword</td>
<td>Encrypted password validated by the PDF Document Engine when it authenticates requests. If security is enabled, only requests providing the correct username and password will be served.</td>
<td>This encrypted password is created by performing Step 4 or Step 5 of the above procedure.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: This parameter is optional unless SOAP security is enabled (see Step 8 above). For more on SOAP security, see this chapter’s section called <em>Enabling Soap Security</em>.</td>
<td></td>
</tr>
</tbody>
</table>
| PrintToPdfLogLevel    | Flag that determines if an Electronic Evidence™ Export Utility log file will be created in the cache directory, and if so whether that file’s logging level will be basic or detailed.                                         | **Valid values:**  
|                       | • 0 (no log file)  
|                       | • 1 (basic logging level)  
|                       | • 2 (detailed logging level)  
<p>|                       | <strong>Default</strong>: 0                                                            |        |</p>
<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>WatermarkOverwriteBehaviour</td>
<td>A flag that determines if a pre-existing watermark in a PDF document is</td>
<td>Valid values:</td>
</tr>
<tr>
<td></td>
<td>overwritten by a new watermark</td>
<td>• 0 (Never overwrite)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1 &lt;Reserved for future use&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 2 (Always overwrite)</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: This parameter is optional.</td>
<td>Default: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>: If you are using the OneSpan Sign Application, the value of</td>
</tr>
<tr>
<td></td>
<td></td>
<td>this parameter must be 2.</td>
</tr>
<tr>
<td>Cryptography/OnlineCert</td>
<td>A flag that determines if the PDF Document Engine will search an online</td>
<td>Valid values:</td>
</tr>
<tr>
<td>Verification</td>
<td>Certificate Revocation List (CRL) to investigate if a selected certificate</td>
<td>• 1 (online search)</td>
</tr>
<tr>
<td></td>
<td>is valid</td>
<td>• 0 (no online search; validity checking will access cached CRLs only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default: 1</td>
</tr>
<tr>
<td>Cryptography/UnknownCertificateValidityWarnings</td>
<td>A flag that determines if a warning will appear when the validity of a selected certificate is unknown</td>
<td>Valid values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• true (warning)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• false (no warning)</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: This parameter is optional.</td>
<td>Default: true</td>
</tr>
<tr>
<td>Cryptography/CheckCertificateStatus</td>
<td>A flag that determines if the applicable certificate will be validated when a signature is applied.</td>
<td>Valid values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1 (validate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0 (don't validate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default: 1</td>
</tr>
</tbody>
</table>
### Cryptography/CrlUrl

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>The proxy URL address through which CRL (Certificate Revocation List) files can be requested.</td>
<td>Valid URL address (including port number) of the proxy</td>
<td>Default: <code>&lt;empty string&gt;</code></td>
</tr>
<tr>
<td>Note: This parameter is optional.</td>
<td>Example: http://[12.48.92.01]:1080</td>
<td>Note: The IP string must be surrounded by square brackets, and the port number must be preceded by a colon. If the port is not specified, it defaults to 1080. If the URL protocol is not specified, it defaults to: http://</td>
</tr>
<tr>
<td>Note: If this parameter is specified, the PDF Document Engine will pass through this proxy en route to the CRL address of the server that stores CRL files. Otherwise, it will go directly to the desired CRL address (as specified in the Signing Certificate).</td>
<td>Note: This feature assumes that the DNS on the proxy’s platform is defined to be the desired secure end-point CRL URL address. You can use that definition to configure the ultimate URL address of the endpoint CRL server.</td>
<td></td>
</tr>
<tr>
<td>Note: This parameter was created to provide additional security. You should specify it if you have a proxy inside your firewall which has the responsibility of handling data outside your firewall.</td>
<td>Note: If you are using HTTPS protocol instead of HTTP, you must also configure the following parameters in this table:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• TimeStampServer.SSLCert</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• TimeStampServer.SSLKey</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• TimeStampServer.SSLPassword</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• TimeStampServer.CA</td>
<td></td>
</tr>
<tr>
<td>However, you cannot use HTTPS with the Time Stamp Server at the same time you use HTTPS with the CrlUrl proxy.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Configuring a PDF Document Engine

### PreferredRevocationCheckingMethod
The method that will be used to retrieve online revocation data about certificates

<table>
<thead>
<tr>
<th>Possible values:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• <strong>CRL</strong> (default) — The PDF Document Engine will initially try to retrieve CRLs for the relevant certificate. If a CRL is not found, the engine will try to retrieve an OCSP response.</td>
</tr>
<tr>
<td>• <strong>OCSP</strong> — The PDF Document Engine will initially try to retrieve OCSP responses for the relevant certificate. If an OCSP response cannot be retrieved, the engine will try to retrieve a CRL.</td>
</tr>
<tr>
<td>• <strong>Optimized</strong> — The PDF Document Engine will use an adaptive strategy to retrieve revocation data. It will choose OCSP or CRL, depending on the context of the certificate-examination operation. In most cases, it will use OCSP for <em>Personal Certificate Signing</em>, and CRL for <em>Click-to-Sign</em>.</td>
</tr>
</tbody>
</table>

**Note:** We highly recommend using the value **Optimized**.

### OCSPResponseValidation
A parameter that specifies how OCSP responses will be validated

<table>
<thead>
<tr>
<th>Possible values:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• <strong>0</strong> — The OCSP responses will not be validated.</td>
</tr>
<tr>
<td>• <strong>1</strong> (default) — A complete validation of OCSP responses will be executed.</td>
</tr>
<tr>
<td>• <strong>WithoutRevocationCheck</strong> — All validation will be performed except examining the revocation status of the certificate used to sign the OCSP response.</td>
</tr>
</tbody>
</table>
### PARAMETER

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI of the server that applies RFC 3161 timestamps to Adobe Signatures</td>
<td>Valid values have the following format: <code>&lt;timestamp_server_url&gt;://&lt;port&gt;/&lt;timestamp_path&gt;</code></td>
</tr>
<tr>
<td><strong>Note:</strong> This parameter is optional. If it is not specified, timestamps will not be applied to Adobe Signatures.</td>
<td></td>
</tr>
</tbody>
</table>
| TimeStampServer/URI | **Note:** If the URL is using HTTPS protocol, you must also configure the following parameters in this table:  
  - `TimeStampServer.SSLCert`
  - `TimeStampServer.SSLKey`
  - `TimeStampServer.SSLPassword`
  - `TimeStampServer.CA` |
| TimeStampServer/CA | Absolute path of the directory that contains the Trusted Certificate Authority files for the SSL certificate  
**Note:** This parameter is required for HTTPS protocol.  
For more on this directory, see Configuring the Trusted CA Folder for HTTPS (page 156). |
| TimeStampServer/SSLCert | Name of the SSL certificate file  
**Note:** This parameter is required for HTTPS protocol. |
|  | The value can be the file’s full path, or just its filename.  
If no path is specified: (1) the software will assume that the file is in the resources folder of the Remote Properties Repository for your configured tenant; (2) you must put the file in that folder. To do so, see Appendix C of the Platform Administrator’s Guide. |
### PARAMETER | DESCRIPTION | VALUES
--- | --- | ---
TimeStampServer/SSLKey | Name of the Private Key file used by the SSL certificate | The value can be the file’s full path, or just its filename.  
**Note:** This parameter is required for HTTPS protocol.  
If no path is specified: (1) the software will assume that the file is in the resources folder of the Remote Properties Repository for your configured tenant; (2) you must put the file in that folder. To do so, see Appendix C of the Platform Administrator’s Guide.

TimeStampServer/SSLPassword | Password of the Private Key file used by the SSL certificate | If no password is specified, the software assumes that this Private Key does not have a password.  
**Note:** This parameter is used with HTTPS protocol.

CoreTrace | Logging level for the "decore.log" and "deservice.log" files created by the PDF Document Engine (for more about these files, see Appendix B) | Valid values:  
- **ERROR** (default): Logs only errors  
- **WARNING**: Logs errors and warnings  
- **INFO**: Logs errors, warnings, and executed PDF Document Engine commands  
- **DEBUG**: Logs errors, warnings, and executed PDF Document Engine commands (including a detailed list of the commands’ parameters and returned data)  
- **DEBUG1**: Logs errors, warnings, executed PDF Document Engine commands (including a detailed list of the commands’ parameters and returned data), and information from the program’s startup that can help debug configuration issues  
**Note:** This parameter is optional.
<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
</table>
| PEMPath   | Full path to the directory that contains all certificates associated with a \*.pem file | The value should be a valid filesystem path.  
Default (non-tenant environment): `<PDF Document Engine deployment directory>/.pem`  
Default (tenant or mixed environment): `<PDF Document Engine deployment directory>/repo/<tenant name>/Resources`.  
Note that the default tenant name is `system`.  
**Note:** This parameter is not present by default in the Admin Console. If you need to configure `PEMPath` in a tenant environment, you must first add it to the PDF Document Engine parameters in Table C-20 of the *Platform Administrator’s Guide* (see that guide’s procedure *Creating/ Editing/Deleting Configuration Properties*). |
TrustedCertificates

Name of the *.pem file that contains all trusted certificates (= "root" certificates). To create the file, see Creating PEM Files from a PFX File in Ch. 5 of the Platform Administrator's Guide.

**Note:** If you're in a non-tenant environment, this file should be in the directory specified by the parameter PEMPath.

**Note:** If you’re in a tenant environment, you must use the Administration Console to: (1) specify this parameter; (2) copy the *.pem file into the Resources directory of each tenant.

**Warning:** Never specify this parameter for the System Tenant.

**Warning:** This *.pem file does not contain the private key. That key is in the *.pem file specified by the parameter CPSFile.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>TrustedCertificates</td>
<td>Name of the *.pem file that contains all trusted certificates (= &quot;root&quot; certificates). To create the file, see Creating PEM Files from a PFX File in Ch. 5 of the Platform Administrator's Guide.</td>
<td>The value should be a valid filename (NOT the full path to the file).</td>
</tr>
</tbody>
</table>
### IntermediateCertificates

**PARAMETER** | **DESCRIPTION** | **VALUES**
--- | --- | ---
IntermediateCertificates | Name of the *.pem file that contains all intermediate certificates (all such certificates should be placed sequentially in this file). To create the file, see Creating PEM Files from a PFX File in Ch. 5 of the Platform Administrator’s Guide. | The value should be a valid filename (*NOT* the full path to the file).  

**Note:** If you’re in a non-tenant environment, this file should be in the directory specified by the parameter PEMPath.  

**Note:** If you’re in a tenant environment, you must use the Administration Console to: (1) specify this parameter; (2) copy the *.pem file into the Resources directory of each tenant.  

**Warning:** Never specify this parameter for the System Tenant.  

**Warning:** This *.pem file does not contain the private key. That key is in the *.pem file specified by the parameter CPSFile.
### PARAMETER | DESCRIPTION | VALUES
---|---|---
SignerCertificate | Name of the *.pem file that contains the certificate for the PDF Document Engine (= the "leaf" certificate). To create the file, see Creating PEM Files from a PFX File in Ch. 5 of the Platform Administrator’s Guide. | The value should be a valid filename (NOT the full path to the file).  
**Note**: If you’re in a non-tenant environment, this file should be in the directory specified by the parameter PEMPath.  
**Note**: If you’re in a tenant environment, you must use the Administration Console to: (1) specify this parameter; (2) copy the *.pem file into the Resources directory of each tenant.  
**Warning**: Never specify this parameter for the System Tenant.  
**Warning**: This *.pem file does not contain the private key. That key is in the *.pem file specified by the parameter CPSFile.  
| Reserved for future use. | The value of this parameter must be 1. Do not modify it.  
RenderActionCompatibilityMode | A flag that determines if the cache directory will create backup copies of one or more original files sent by the PDF Document Engine's client. The backups are created mostly for debugging purposes. | Valid values:  
• 0 (no backup file)  
• 1 (backup file or files)  
Default: 0  
BackupOriginalFile
## ImageBitDepth

Bit depth of the image of a rendered PDF document if the image format is **PNG**, and if the color model is **GRAYSCALE** or **RGB**. This parameter is ignored if the image format is not **PNG** or if the color model is **MONO** (Monochrome).

Valid values:
- 2
- 4
- 8 (default)

**Note**: If no value is specified or if the specified value is invalid, *OneSpan Sign* uses the default value.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageBitDepth</td>
<td><strong>Bit depth of the image of a rendered PDF document if the image format is PNG, and if the color model is GRAYSCALE or RGB. This parameter is ignored if the image format is not PNG or if the color model is MONO (Monochrome).</strong></td>
<td>Valid values: 2, 4, 8 (default). <strong>Note</strong>: If no value is specified or if the specified value is invalid, <em>OneSpan Sign</em> uses the default value.</td>
</tr>
</tbody>
</table>

## XvfbDisplay

Value of the **X Virtual Frame Buffer** used by the PDF Document Engine to render HTML files to PDF

The string that replaces the placeholder `<value>` in the following expression:

```
Xvfb <value> -auth /etc/X99.hosts -screen 0 1x1x16 &
```

Default = :99

**Note**: If this value is defined, it must be the same as that in the file "rc.local", which is in the directory "/etc/rc.d" (see the **Prerequisites of the procedure Installing a New PDF Document Engine on Linux**).
<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
</table>
| ResourcePath   | Full path to the directory that contains the Resource sub-directories CMap, Color, Font, and Unicode. | Note: In a tenant environment, this parameter can be set only by the System Tenant. In a non-tenant environment, it can be set only via the `DocumentEngineSettings.props` file (as usual). The value must be a valid filesystem path. Default: The Resource directory associated with the installation of the PDF Document Engine. For example:  
  - On Linux: `<DE deployment directory>/silanis/Resource`  
  - On Windows: `<DE deployment directory>\services\DEService\Resource` |
| PreservePDFAConformance | A flag that determines how the PDF Document Engine will execute actions on a PDF file that conforms to one of the following PDF Archive standards: | Valid values:  
  - 1 — Any action applied by the engine to a PDF/A-conformant file will preserve the same level of conformance. Note: If the action is `getElectronicEvidence` (input file = an HTML file), the output PDF will be PDF/A-conformant if the invoked PDF template is PDF/A-conformant.  
  - 0 (default) — Any action applied by the engine to a PDF/A-conformant file will not preserve the same level of conformance. |  
  - PDF/A-2b  
  - PDF/A-2u  
  - PDF/A-3b  
  - PDF/A-3u  

The next column refers to such files as "PDF/A-conformant".
### PARAMETER | DESCRIPTION | VALUES
--- | --- | ---
PrintToPdfAllowSelfSignedCertificate | A flag that determines if an HTTPS call encountered inside an HTML file for the production of *Electronic Evidence* will be granted access to a server that uses a self-signed certificate. **Note:** This flag is used only in producing *Electronic Evidence* via the PDF Document Engine's server. For example, it is not used by the Core Component. | Valid values:  
• 1 — Any HTTPS call encountered inside an HTML file for the production of *Electronic Evidence* will be granted access to any server that uses a self-signed certificate.  
• 0 (default) — Any HTTPS call encountered inside an HTML file for the production of *Electronic Evidence* will be denied access to any server that uses a self-signed certificate.  

LogFilter | A parameter that can filter confidential data out of the "decore.log" file if CoreTrace = DEBUG or DEBUG1 | Possible values: A comma-separated list of PDF Document Engine parameters  
No parameter in this list will appear in the "decore.log" file. Instead, that file will display several asterisks (per parameter).  

Default: `<empty string>`  
Thus the default is *not* to filter confidential data.  
If you want to filter out *all* confidential data, put all of the following parameters in the list:  
• user_ID  
• location  
• reason  
• comment  
• label_text  
• audit_trail_user_name  
• proc_hash  
• signature_data  
• token

DecoreSize | Desired maximum size (in KB) of the "decore.log" and "deservice.log" files before those files "roll over" into a new file (for more on this maximum size, see Appendix B) | Any integer value is acceptable.  
Default: 1000
<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogMaxCount</td>
<td>Desired maximum number of old &quot;decore.log&quot; files and old &quot;deservice.log&quot; files created during log-file rotations</td>
<td>Any integer value is acceptable. Default: 20 &lt;br&gt;&lt;br&gt;Note: The value 0 corresponds to unlimited number of old files — i.e., even the oldest files will not be deleted during a log rotation.</td>
</tr>
<tr>
<td>TraceSOAP</td>
<td>Flag that determines if the PDF Document Engine will trace SOAP requests and responses</td>
<td>Valid values: &lt;br&gt;&lt;br&gt;• 0 (false; default) &lt;br&gt;&lt;br&gt;• 1 (true) &lt;br&gt;&lt;br&gt;Note: The value 1 should be used only for diagnostic purposes.</td>
</tr>
<tr>
<td>RepoRefresh</td>
<td>Interval (in seconds) after which the repository for PDF Document Engine properties will be refreshed</td>
<td>Default: 86400 (= 1 day) &lt;br&gt;&lt;br&gt;Note: If you deliberately set this parameter to zero, the repository for PDF Document Engine properties will be read on startup, but it will never be refreshed while the PDF Document Engine is running. &lt;br&gt;&lt;br&gt;Note: If the repository for PDF Document Engine properties cannot be refreshed, the existing repository will be reused.</td>
</tr>
<tr>
<td>FlattenFlag</td>
<td>Flag that determines if &quot;flatten&quot; actions will flatten all fields and all annotations, or only all fields</td>
<td>Valid values: &lt;br&gt;&lt;br&gt;• 0 (false) — All fields will be flattened, but no annotations. &lt;br&gt;&lt;br&gt;• 1 (true; default) — All fields and all annotations will be flattened.</td>
</tr>
<tr>
<td>FlattenURL</td>
<td>Flag that determines if &quot;flatten&quot; actions will render non-functional the URLs within flattened PDFs</td>
<td>Valid values: &lt;br&gt;&lt;br&gt;• 0 (false; default) — No URLs will be flattened. &lt;br&gt;&lt;br&gt;• 1 (true) — All URLs will be flattened.</td>
</tr>
<tr>
<td>PARAMETER</td>
<td>DESCRIPTION</td>
<td>VALUES</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PrintToPdfSingleHTML</td>
<td>Parameter that determines how the Electronic Evidence™ Export Utility will be called when it operates on an e-signature process that contains multiple HTML files</td>
<td>Valid values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0 — The utility will be called <em>only once</em> for all HTML files in the series. This alternative may consume more memory, but it will take less time to execute.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1 (default) — The utility will be called once for each HTML file in the series. This alternative requires less memory, but will take longer to execute.</td>
</tr>
<tr>
<td>CommunicationPrivateKey</td>
<td>The full path of a *.pem file that contains the private key that will be used to sign messages sent from the PDF Document Engine to the Personal Certificate Client</td>
<td>These three parameters are designed to be used together, and are relevant only if your solution uses the Personal Certificate Client.</td>
</tr>
<tr>
<td>CommunicationPrivateKeyPassword</td>
<td>Encrypted password of the <code>CommunicationPrivateKey</code> in the previous row</td>
<td>If you do <em>not</em> specify these parameters, the PDF Document Engine will sign messages sent from the PDF Document Engine to the Personal Certificate Client using: (1) the private key specified by the parameter <code>CPSFile</code>; (2) the certificate specified by the parameter <code>SignerCertificate</code>.</td>
</tr>
<tr>
<td>CommunicationCertificate</td>
<td>Name of the *.pem file that contains the certificate corresponding to the private key specified by the parameter <code>CommunicationPrivateKey</code></td>
<td>Specify these three parameters if and only if you want to use another private key and certificate to sign such messages.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> If you are using an Hardware Security Module (HSM) to store the private key for signing PDFs, you must specify these three parameters.</td>
</tr>
</tbody>
</table>
### CriticalErrors

A string that defines certain errors as **critical**.

If a PDF Document Engine process encounters a "critical error", then: (1) that process will exit immediately; (2) if necessary (depending on the current load), Apache will spawn a fresh process in its place.

**Note**: This parameter was designed for use with a Linux installation of the PDF Document Engine. Using this parameter on a Windows installation is not recommended.

- **VALUES**: A critical error is identified by its error code and the content of its error message. Valid values are therefore `error-code;error-message` pairs, with the two parts separated by a semi-colon. If an error is to be considered critical regardless of its message, an empty string must nonetheless be specified for its message entry. To disable the processing of critical errors, specify `-1` as the first critical error code (e.g., `CriticalErrors = -1`). The value `-1` is this parameter’s default value.

**Example:**

The following parameter assignment identifies two types of critical errors, with codes 66 and 67 (separated by a semi-colon):

```
CriticalErrors=66;8009108D;67;;
```

In this example: (1) an error with code 66 will be considered critical if its message contains the string `8009108D`; (2) an error with code 67 will always be considered critical, irrespective of its message.

### RSALabel

Attribute that defines the label of the key handle for the private key associated with an HSM.

**Note**: This label is not used by Gemalto’s HSM.

- **VALUES**: A user-defined string

**Note**: This parameter is defined when the HSM Client software is configured.
<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutographFrame</td>
<td>A parameter that determines the appearance of a handwritten signature in a</td>
<td>• FIT (default) — The handwritten signature will be: (1) centered in the Signature Block (both</td>
</tr>
<tr>
<td></td>
<td>Signature Block</td>
<td>horizontally and vertically); (2) stretched to fit the entire block.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ACTUALSIZE — The entire capture area of the handwritten signature will be inserted in the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Signature Block.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>There are four valid values. Here are the first three (with illustrative logos):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• LIGHT (default):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• LIGHTER:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SOLID:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: The above values assume that the Signature Block’s background color is transparent or</td>
</tr>
<tr>
<td></td>
<td></td>
<td>white. If it isn’t, OneSpan Sign automatically uses a more visible logo. If the background is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>green, for example, the logo appears as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The fourth valid value is a PDF file that contains a customized logo. For more on this option,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>see the footnotes below.</td>
</tr>
</tbody>
</table>

a. This value has the following format: file://<transparent-background-logo-file>;<<non-transparent-background-logo-file>. Here: (1) <transparent-background-logo-file> is the filename or complete path of the PDF to be used if the Signature Block’s background is transparent or white; (2) <non-transparent-background-logo-file> is the filename or complete path of the PDF to be used if the Signature Block’s background is not transparent or white. Including <non-transparent-background-logo-file> is optional. If it is not specified, the <transparent-background-logo-file> PDF will be used, regardless of the Signature Block’s background.
Configuring a PDF Document Engine

b. If you specify only the PDF’s filename rather than its complete path, the PDF must be provided as follows: (1) in a "tenant environment", the PDF must be specified as a tenant resource via the Admin Console; (2) in a "non-tenant environment", the PDF file must be in the directory <DE deployment directory>/logos/.

c. Example 1: SilanisLogo = file://drawing1.pdf. Here the system always uses drawing1.pdf (provided as a resource file in a tenant environment, or provided in the folder <DE deployment directory>/logos/ in a non-tenant environment). Example 2: SilanisLogo = file://drawing1.pdf;/home/silanis/opaque.pdf. Here: (1) if the Signature Block’s background is transparent or white, the system uses drawing1.pdf (provided as a resource, assuming a tenant environment); (2) otherwise, the system uses /home/silanis/opaque.pdf (located on the machine where the PDF Document Engine is installed).

d. (1) The PDF file version must be 1.4 to 1.7. (2) The PDF/A format is preferred. (3) The PDF file should not contain raster images. Only vector images are allowed. (4) Using text is not recommended. It is preferable to convert any text to a vector image. (5) If text content cannot be avoided, the fonts needed to properly render the text should be embedded in the PDF. (6) The document should not contain any active contents (e.g., JavaScript, Movie Annotation, XFA). (7) The PDF’s file size should be as small as possible, since the image will be embedded in all signed documents.

**Configuring the Trusted CA Folder for HTTPS**

If you are using HTTPS protocol with the Time Stamp Server or the CrlUrl proxy described in Table 11-2, you must first configure the directory for your Trusted Certificate Authority.

**To configure the Trusted CA folder on Linux:**

1. If you don’t already have it, install openssl-perl using the following command:

   ```bash
   yum install openssl-perl
   ```

2. Place the Trusted Certificate Authority files in a designated directory `<dir path>`.

3. Run the following command:

   ```bash
   c_rehash <dir path>
   ```

   This procedure ends here.

**To configure the Trusted CA folder on Windows:**

1. Install a binary version of openssl on your Operating System.

2. In your global settings for the PATH variable, add the path to the location of the file openssl.exe.

3. Put all relevant Trusted Certificate Authority files in your Trusted Certificate Authority folder, and enter that folder.

4. For each CA file, execute the following command:

   ```bash
   openssl x509 -hash -in <Trusted CA file>
   ```

5. For each CA file, record the hash value, `<hash>`, that appears on the console (it’s the first line you see from the openssl command).

6. For each CA file, execute the following command:

   ```bash
   mklink <hash>.0 <Trusted CA file>
   ```

   This procedure ends here.
Enabling SOAP Security

You may want to enhance the security of SOAP communications between the PDF Document Engine and the Core Component by requiring that the latter provide valid credentials with its SOAP requests. If so, you should “enable SOAP security” by taking the steps in the following procedure.

To enable SOAP security:

1. If you are using JBoss or Wildfly, perform Step 10 of Chapter 8’s procedure Installing the Core Component on JBoss or Wildfly.

2. If you are in a non-tenant environment, specify the parameters RampartUsername and RampartPassword in Table 11-2, using the file DocumentEngineSettings.props.

3. If you are in a tenant environment, specify the parameters RampartUsername and RampartPassword in Table 11-2 using the Administration Console. You must: (1) do this separately for the System Tenant and the relevant other tenant; (2) give both the System Tenant and the other tenant the same RampartUsername and same RampartPassword.

4. Perform Step 8 of this chapter’s procedure Configuring the PDF Document Engine.


⚠️ SOAP security will not be completely enabled until all the above steps have been taken.

This procedure ends here.

Uninstalling PDF Document Engines

⚠️ Don’t forget to backup your PEM files to a safe location before you uninstall. When you restore any of those files, make sure the correct ones are being restored.

⚠️ If you want to uninstall more than one PDF Document Engine, perform the following procedure on each engine.

To uninstall a PDF Document Engine, do the following on the machine where it is installed:

1. If Apache or IIS HTTP Server is running, the superuser should stop it.

2. If the PDF Document Engine is installed on Linux, the superuser should do the following:
a. From within the install directory, run the following command (replacing the placeholder in angle brackets):

```
./de_uninstall.sh <DE deployment directory>
```

b. If you are using RHEL/CentOS 6, undo any modifications to the file /etc/init.d/httpd done during installation.

c. Remove the file silanis.conf from the conf.d sub-folder of your Apache installation (default = /etc/httpd/conf.d for a normal yum installation).

d. If you are using RHEL/CentOS 6, delete the file /etc/init/DEMultitenancy.conf.

e. If you are using RHEL/CentOS 7, delete the files DEHttpd.service and DEMultitenancy.service from the services directory of the Operating System (e.g., /usr/lib/systemd/system).

If you want the PDF Document Engine to be uninstalled by the application user, see Appendix J.

3. If the PDF Document Engine was installed on Windows using IIS:

a. Open the IIS configuration file "applicationHost.config" in the directory "%systemroot%/Windows/System32/inetsrv/config".

b. Change the tag `<serverRuntime uploadReadAheadSize="52428800" />

   back to `<serverRuntime />

   .

c. Save and close the configuration file.

d. Remove the DLL library "mod_axis2_IIS.dll" as a Wildcard Script Map by doing the following:

   A. Click Start > Administrative Tools > Internet Information Services (IIS) Manager. The corresponding window appears.

   B. Expanding the tree in the left pane (Connections), under Sites click [IIS Website].

   C. In the right pane, right-click Handler Mappings > Open Feature.

   D. In the Handler Mappings pane, right-click the name assigned during the installation (e.g., axis), and select Remove.

   E. In the left pane (Connections), click [IIS Website].

   F. In the rightmost pane (Actions), click Start.

4. If the PDF Document Engine is installed on Windows, uninstall it using a method supported by your Windows Operating System. You can use either of the following general methods:

   • Run the PDF Document Engine setup.exe program, and select its Uninstall option. You can optionally uninstall Apache as well as the PDF Document Engine.
• Use the *Add/Remove Programs* facility. This will remove the PDF Document Engine. If you also want to remove Apache, you need to use this facility again.

  This procedure ends here.
PART IV: OPTIONAL PLATFORM SOFTWARE

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12: Deploying e-Witness</td>
<td>161</td>
</tr>
<tr>
<td>13: Deploying the Archival Module</td>
<td>174</td>
</tr>
<tr>
<td>14: Working with the e-Vault Manager</td>
<td>178</td>
</tr>
</tbody>
</table>
CHAPTER 12: Deploying e-Witness

e-Witness is a OneSpan Sign module that enables authorized parties to find, review, and verify a stored e-signature process. Those parties are enabled to view all of the screens presented to the signers as they went through the signing process.

Before OneSpan Sign 5.1, e-Witness was known as the Process Reviewer. All occurrences of the term Process Reviewer in this guide or in the product itself refer to e-Witness.

The Electronic Evidence™ Export Utility is installed together with e-Witness, and is enabled through the use of a license. For a specified e-signature process, this utility prints into a PDF file or a Web Archive HTML file all the screens that would be displayed by e-Witness. The output content appears in the same order as it would in e-Witness screens.

The exported HTML format is HTML 4.01 ISO/IEC 15445; ISO 8879. Note that Internet Explorer does not fully implement this format.

This chapter describes:

• Installing e-Witness (page 161)
• Uninstalling e-Witness (page 169)

Installing e-Witness

To install e-Witness:

1. Ensure that all General Prerequisites (page 161) are satisfied.

2. Perform one of the following procedures (depending on your Application Server):
   • Installing e-Witness on WebSphere (page 164)
   • Installing e-Witness on JBoss or Wildfly (page 166)
   • Installing e-Witness on WebLogic (page 168)

General Prerequisites

Whatever Application Server you use, your installation of e-Witness should begin by ensuring that the following prerequisites are met.

• Your environment meets all relevant requirements in Chapter 2.
• You have performed the relevant procedure from Preparing an App Server Environment (page 312).
• The following procedures have been performed:
   • Installing the Admin Console (page 65)
   • Installing the Transaction Status API (see the Transaction Status API)
Deployment Guide provided by OneSpan

- You have created a user with permissions to use e-Witness. For details on how to do so, see the Platform Administrator’s Guide.
- The PDF Document Engine is installed.

If you want to use the Electronic Evidence™ Export Utility, we highly recommend that the installed PDF Document Engine be dedicated to e-Witness (for performance reasons). To do so, you must specify appropriate values for two parameters in Table C-10 of the Platform Administrator’s Guide (namely, documentEngineURL and documentEngineURL.force).

The remaining prerequisites are relevant only if you want to use the Electronic Evidence™ Export Utility.

- If you want to use the Electronic Evidence™ Export Utility, you have correctly installed the license file that permits its use. If you are in a clustered environment, that file must be present on every Application Server.
- If you want to use the Electronic Evidence™ Export Utility, you have acquired a valid Java keystore. To create a keystore for a production environment, see Appendix E.
- Optional: If you want to use the Electronic Evidence™ Export Utility and you are using OneSpan Sign in a clustered Application Server environment, you might want to store the PDF files and Web Archive HTML files created by the Electronic Evidence™ Export Utility in a directory that is shared and accessible by each Application Server in the cluster. If you don’t do so, each member of the cluster will save PDF files and Web Archive HTML files in its own directory. Note that this directory is also used to save the statuses of export processes.
- If you want to use the Electronic Evidence™ Export Utility and the PDF Document Engine is on Linux, the user who runs Apache has specified the DISPLAY environment variable appropriately (DISPLAY = ‘99).
- If you want to use the Electronic Evidence™ Export Utility and the PDF Document Engine is on Windows:
  a. A user has been created and assigned to the PDF Document Engine’s Apache or IIS HTTP Server, thereby enabling them to run it. If you are using IIS, this is the [IIS User] that was created and assigned to the IIS HTTP Server in Chapter 11.

  This user must have READ/WRITE access to the CachePath parameter in Table 11-2.

  b. That user has created a default printer, and has associated it with their user account (if no printer is associated with that user, a printer must be added via the Control Panel). The PDF Document

© 2020 - OneSpan North America Inc.      DG-OSS-72 - All rights reserved.
Engine uses that printer to convert HTML content into PDF content. Therefore:

A. Ensure that there is a valid printer driver by verifying that at least one printer is listed under *Printers*.

If Apache has been configured to run as a Windows service or if you are using IIS, the default printer must be local.

If Apache has been configured to run as an application, the default printer may be either local or network.

B. If the desired printer is not listed, use Admin privileges to add the required printer driver from the *Printers* list. In most circumstances, the relevant printer should be local.

Rendering performance may be affected by the quality of the printer driver you select. If more than one printer is available, you might want to run tests to see which one is best for the *Electronic Evidence* Export Utility.

C. Ensure that the *Electronic Evidence* Export Utility user has access permission for the printer.

c. The printer used by the *Electronic Evidence* Export Utility supports your preferred paper size (letter, legal, or A4).

d. If the PDF Document Engine uses IIS, the *Electronic Evidence* Export Utility user must have a normal account for the IIS Application Pool. Therefore:

A. Start the *IIS Manager*.

B. In the left pane, click *Application Pools*. The middle pane changes.

C. In the middle pane, right-click *DefaultAppPool*, and select *Advanced Settings*. The *Advanced Settings* window appears.

D. In the *Process Model* area, select *Identity* and click *OK*. The *Application Pool Identity* dialog box appears.

E. Select *Custom account*, and click *Set*...A *Set Credentials* dialog box appears.

You must select *Custom account*. Do NOT select *Built-in account*.

F. Edit the user's credentials. They must be the same as the credentials for the [*IIS User*] created in Chapter 11.

G. To finish the configuration, click *OK*.
Installing e-Witness on WebSphere

This section describes how to *manually* install e-Witness on WebSphere. Chapter 3 describes an alternate, *automated* way of doing so.

Prerequisites

- All General Prerequisites (page 161) are satisfied.

If you are in a clustered environment, every file mentioned in the following procedure must either be present on every node in the cluster, or must be on a network drive that can be accessed by all nodes in the cluster.

Action

To install e-Witness on WebSphere:

1. Configure the Application Server with the following Java Virtual Machine (JVM) system property:
   - `USER_SERVICE_URL = https://<User_Master_Server>:<port>/user/rest/v1`

2. If you have *not* installed the Core Component or if you want to install e-Witness on a different server from the Core Component:
   a. On the Application Server host, create a directory that will serve as the OneSpan Sign home directory.
   b. Into this `[eSL_HOME]` directory, copy the following file:
      - `silanisLicense.xml`
   c. Configure the Application Server with the following Java Virtual Machine (JVM) system properties:
      - `AWS_HOME = [eSL_HOME]`
      - `RESOURCE_SERVER_LOCATION = https://<Resource Manager Server>:<port>/resources-master/storage`
   d. If you want to change the local cache location (which by default is the system-dependent temporary-file directory, together with the suffix `esep`), configure the Application Server with the following Java Virtual Machine (JVM) system property:
      - `RESOURCE_LOCAL_CACHE = <path to local cache>`
      The directory of this local cache should never be auto-cleaned. Yet on RHEL/CentOS, the system-dependent temporary-file directory is auto-cleaned by default.
   e. If you want to change the default expiration time of the local cache (which is 900 seconds), configure the Application Server with the following Java Virtual Machine (JVM) system property:
      - `CACHE_LIFETIME_SECONDS = <N>`
Installing e-Witness on WebSphere

f. Create a Data Source for e-Witness that points to the Core Database (for details, see Step 11 in Chapter 8).

3. If you have installed the Core Component, a Data Source for the Core Database was created in Chapter 8. By default, e-Witness will use that Data Source. However, you can specify alternative Data Sources if you wish (see Appendix A of the Platform Administrator's Guide).

4. If you are using WebSphere SIB default messaging, create the following destination within the bus used by the Core Component:
   - PrintToPdfQueueDestination

5. To complete the configuration of JMS resources:
   a. Using the parameters in Table 12-1, in the Application Server create a persistent JMS queue called PrintToPdfQueue.

   For the above queue: (1) from the WebSphere Administration Console, go to Resources > JMS > Queues > Queue name > Advanced properties; (2) select the checkbox Append RFH version 2 headers to messages sent to this destination Message.

   b. Create the following activation specification:
      - Name: ActivationSpecPrintToPdf
      - JNDI name: jms/ActivationSpecPrintToPdf

6. Inside the Application Server, deploy the file "eSL-eWitness-WebSphere.ear" provided by OneSpan.

7. Configure HTTPS access to e-Witness.

8. Restart the Application Server that hosts the AWSReviewer.

9. Verify the above installation by using a browser to go to the following URL:
   - https://<server>:<port>/evidence

   Here <server> denotes the server where e-Witness is installed, and <port> denotes the port it uses. The logon page of e-Witness should appear.

   You must also configure e-Witness separately for each tenant using the Platform Administrator's Guide (see Tables C-9, C-10, C-11 and C-12 of that guide).

   This procedure ends here.

NOTE: The above steps complete the installation. However, you won't be able to access e-Witness functionality until you perform the procedure Accessing e-Witness in Chapter 2 of the e-Witness User's Guide. This will enable you to access all e-Witness functionality via a GUI (including exporting Electronic Evidence™ to a PDF file or a Web Archive HTML file).
Installing e-Witness on JBoss or Wildfly

This section describes how to manually install e-Witness on JBoss or Wildfly. Chapter 3 describes an alternate, automated way of doing so.

Prerequisites

- All General Prerequisites (page 161) are satisfied.

Action

To install e-Witness on JBoss or Wildfly:

1. Configure the Application Server with the following Java Virtual Machine (JVM) system property:
   - USER_SERVICE_URL = https://<User_Master_Server>:<port>/user/rest/v1

2. If you have not installed the Core Component or if you want to install e-Witness on a different server from the Core Component:
   a. On the Application Server host, create a directory that will serve as the OneSpan Sign home directory.
   b. Into this [eSL_HOME] directory, copy the following file:
      - silanisLicense.xml
   c. Configure the Application Server with the following Java Virtual Machine (JVM) system properties:
      - AWS_HOME = [eSL_HOME]
      - RESOURCE_SERVER_LOCATION = https://<Resource Manager Server>:<port>/resources-master/storage

Table 12-1: PrintToPdf Queue parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the PrintToPdf Queue</td>
<td>PrintToPdfQueue</td>
</tr>
<tr>
<td>WebSphere calls this parameter the Base queue name.</td>
<td></td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
<td>APPROVEIT.PRINTTOPDF.QUEUE</td>
</tr>
<tr>
<td>JNDI name associated with the PrintToPdf Queue</td>
<td>The JNDI name must contain the queue name as follows. If the queue name is &quot;PrintToPdfQueue&quot;, the JNDI name must be &quot;jms/queue/PrintToPdfQueue&quot;.</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
<td>Persistent</td>
</tr>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
<td>2</td>
</tr>
<tr>
<td>WebSphere MQ and WebSphere call this parameter the Backout threshold.</td>
<td></td>
</tr>
</tbody>
</table>
d. If you want to change the local cache location (which by default is the system-dependent temporary-file directory, together with the suffix `esep`), configure the Application Server with the following Java Virtual Machine (JVM) system property:

   • `RESOURCE_LOCAL_CACHE = <path to local cache>`

   The directory of this local cache should never be auto-cleaned. Yet on RHEL/CentOS, the system-dependent temporary-file directory is auto-cleaned by default.

e. If you want to change the default expiration time of the local cache (which is 900 seconds), configure the Application Server with the following Java Virtual Machine (JVM) system property:

   • `CACHE_LIFETIME_SECONDS = <N>`

f. Create a Data Source for e-Witness that points to the Core Database (for details, see Step 10 in Chapter 8).

3. If you have installed the Core Component, a Data Source for the Core Database was created in Chapter 8. By default, e-Witness will use that Data Source. However, you can specify alternative Data Sources if you wish (see Appendix A of the Platform Administrator’s Guide).

4. Use the parameters in Table 12-1 to create a persistent JMS queue called `PrintToPdfQueue`.

5. Inside the Application Server, deploy the file "`eSL-eWitness.ear`".

6. Configure HTTPS access to e-Witness.

7. Restart the Application Server that hosts the AWSReviewer.

8. Verify the above installation by using a browser to go to the following URL:

   • `https://<server>:<port>/evidence`

   Here `<server>` denotes the server where e-Witness is installed, and `<port>` denotes the port it uses. The logon page of e-Witness should appear.

   You must also configure e-Witness separately for each tenant using the Platform Administrator’s Guide (see Tables C-9, C-10, C-11 and C-12 of that guide).

   This procedure ends here.

**NOTE:** The above steps complete the installation. However, you won’t be able to access e-Witness functionality until you perform the procedure Accessing e-Witness in Chapter 2 of the e-Witness User’s Guide. This will enable you to access all e-Witness functionality via a GUI (including exporting Electronic Evidence™ to a PDF file or a Web Archive HTML file).
Installing e-Witness on WebLogic

Prerequisites

• All General Prerequisites (page 161) are satisfied.

⚠️ If you are in a clustered environment, every file mentioned in the following procedure must either be present on every node in the cluster, or must be on a network drive that can be accessed by all nodes in the cluster.

Action

To install e-Witness on WebLogic:

1. Configure the Application Server with the following Java Virtual Machine (JVM) system property:
   • USER_SERVICE_URL = https://<User_Master_Server>:<port>/user/rest/v1

2. If you have not installed the Core Component or if you want to install e-Witness on a different server from the Core Component:
   a. On the Application Server host, create a directory that will serve as the OneSpan Sign home directory.
   b. Into this [eSL_HOME] directory, copy the following file:
      • silanisLicense.xml
   c. Configure the Application Server with the following Java Virtual Machine (JVM) system properties:
      • AWS_HOME = [eSL_HOME]
      • RESOURCE_SERVER_LOCATION = https://<Resource Manager Server>:<port>/resources-master/storage
      • Javax.wsdl.factory.WSDLFactory = com.ibm.wsdl.factory.WSDLFactoryImpl
   d. If you want to change the local cache location (which by default is the system-dependent temporary-file directory, together with the suffix esep), configure the Application Server with the following Java Virtual Machine (JVM) system property:
      • RESOURCE_LOCAL_CACHE = <path to local cache>

⚠️ The directory of this local cache should never be auto-cleaned. Yet on RHEL/CentOS, the system-dependent temporary-file directory is auto-cleaned by default.

e. If you want to change the default expiration time of the local cache (which is 900 seconds), configure the Application Server with the following Java Virtual Machine (JVM) system property:
   • CACHE_LIFETIME_SECONDS = <N>
f. Create a Data Source for e-Witness that points to the Core Database (for details, see Step 10 in Chapter 8).

3. If you have installed the Core Component, a Data Source for the Core Database was created in Chapter 8. By default, e-Witness will use that Data Source. However, you can specify alternative Data Sources if you wish (see Appendix A of the Platform Administrator’s Guide).

4. Use the parameters in Table 12-1 to create a persistent JMS queue called PrintToPdfQueue.

5. Inside the Application Server, deploy the file "eSL-eWitness.ear".

6. Configure HTTPS access to e-Witness.

7. Restart the Application Server that hosts the AWSReviewer.

8. Verify the above installation by using a browser to go to the following URL:
   
   • https://<server>:<port>/evidence

   Here <server> denotes the server where e-Witness is installed, and <port> denotes the port it uses. The logon page of e-Witness should appear.

   You must also configure e-Witness separately for each tenant using the Platform Administrator’s Guide (see Tables C-9, C-10, C-11 and C-12 of that guide).

This procedure ends here.

NOTE: The above steps complete the installation. However, you won’t be able to access e-Witness functionality until you perform the procedure Accessing e-Witness in Chapter 2 of the e-Witness User’s Guide. This will enable you to access all e-Witness functionality via a GUI (including exporting Electronic Evidence™ to a PDF file or a Web Archive HTML file).

Uninstalling e-Witness

To uninstall e-Witness, you need perform only one of the following procedures (depending on your Application Server):

• Uninstalling e-Witness on WebSphere (page 170)
• Uninstalling e-Witness on JBoss or Wildfly (page 171)
• Uninstalling e-Witness on WebLogic (page 172)
Uninstalling e-Witness on WebSphere

This section describes how to manually uninstall e-Witness on WebSphere. Chapter 3 describes an alternate, automated way of doing so.

To uninstall e-Witness on WebSphere:

1. Inside the Application Server, uninstall the "eSL eWitness" application.

2. If you installed the Electronic Evidence™ Export Utility, delete the following files:
   - template_legal.pdf
   - template_letter.pdf
   - template_a4.pdf
   - template.html

3. If you installed the Electronic Evidence™ Export Utility:
   a. Delete the activation specification created during the installation of the Electronic Evidence™ Export Utility.
   b. Delete the JMS queue PrintToPdfQueue.

4. If you used WebSphere SIB default messaging, and installed the Electronic Evidence™ Export Utility, remove the bus destination PrintToPdfQueueDestination.

5. If you used WebSphere MQ messaging, and installed the Electronic Evidence™ Export Utility, on the MQ server delete the queue APPROVEIT.PRINTTOPDF.QUEUE.

6. Delete the Data Source for e-Witness.

7. Delete the JDBC Provider for e-Witness (provided it’s not being used by any other application).

8. If you copied any JDBC driver files when e-Witness was installed, delete them (provided they’re not being used by any other application).

9. Unset the generic JVM arguments that were set when e-Witness was installed.

10. Perform the following related steps (if you are using a cluster, perform these steps on each Application Server host):
    a. Make a backup copy of the directory "[eSL_Home]/reviewer".

    ! If you do not have a backup copy of this directory, you will permanently lose all its data.

    b. Delete the directory "[eSL_Home]/reviewer" and all its contents.
11. If you created a directory to be shared by all Application Servers in a cluster for the Electronic Evidence™ Export Utility (see the General Prerequisites), perform the following related steps:
   a. Make a backup copy of that directory.

   ![Warning] If you do not have a backup copy of this directory, you will permanently lose all its data.

   b. Delete that directory.

12. Restart the Application Server.

13. If you installed the Audit Service when you installed e-Witness, perform the procedure Uninstalling the Audit Service (page 56).

   This procedure ends here.

Uninstalling e-Witness on JBoss or Wildfly

This section describes how to manually uninstall e-Witness on JBoss or Wildfly. Chapter 3 describes an alternate, automated way of doing so.

To uninstall e-Witness on JBoss or Wildfly:

1. Inside the Application Server delete the file "eSL-eWitness.ear".

2. If you installed the Electronic Evidence™ Export Utility, delete the following files:
   - template_legal.pdf
   - template_letter.pdf
   - template_a4.pdf
   - template.html

3. If you installed the Electronic Evidence™ Export Utility, delete the JMS queue PrintToPdfQueue.

4. Delete the Data Source for e-Witness.

5. Delete the JDBC Provider for e-Witness (provided it’s not being used by any other application).

6. If you copied any JDBC driver files when e-Witness was installed, delete them (provided they’re not being used by any other application).

7. Unset the generic JVM arguments that were set when e-Witness was installed.
8. Perform the following related steps on the Application Server host):
   a. Make a backup copy of the directory "+[eSL_Home]/reviewer".

   ! If you do not have a backup copy of this directory, you will permanently lose all its data.

   b. Delete the directory "+[eSL_Home]/reviewer" and all its contents.

9. Restart the Application Server.

10. If you installed the Audit Service when you installed e-Witness, perform the procedure Uninstalling the Audit Service (page 56).

    This procedure ends here.

**Uninstalling e-Witness on WebLogic**

To uninstall e-Witness on WebLogic:

1. Inside the Application Server, uninstall the "eSL eWitness" application.

2. If you installed the *Electronic EvidenceTM Export Utility*, delete the following files:
   - template_legal.pdf
   - template_letter.pdf
   - template_a4.pdf
   - template.html

3. If you installed the *Electronic EvidenceTM Export Utility*, delete the JMS queue PrintToPdfQueue.

4. Delete the Data Source for e-Witness.

5. Delete the JDBC Provider for e-Witness (provided it’s not being used by any other application).

6. If you were *not* using an Oracle database when e-Witness was installed, and if you copied any JDBC driver files at that time, delete them (provided they’re not being used by any other application).

7. Unset the generic JVM arguments that were set when e-Witness was installed.

8. Perform the following related steps (if you are using a cluster, perform these steps on each Application Server host):
   a. Make a backup copy of the directory "+[eSL_Home]/reviewer".

      ! If you do not have a backup copy of this directory, you will permanently lose all its data.

   b. Delete the directory "+[eSL_Home]/reviewer" and all its contents.
9. If you created a directory to be shared by all Application Servers in a cluster for the Electronic Evidence™ Export Utility (see the General Prerequisites), perform the following related steps:
   a. Make a backup copy of that directory.

   ▶️ If you do not have a backup copy of this directory, you will permanently lose all its data.

   b. Delete that directory.

10. Restart the Application Server.

11. If you installed the Audit Service when you installed e-Witness, perform the procedure Uninstalling the Audit Service (page 56).

    This procedure ends here.
CHAPTER 13: Deploying the Archival Module

The Archival Module is a component of OneSpan Sign that enables administrators to archive data from and restore data to the OneSpan Sign Core Database.

This chapter describes:

- Installing the Archival Module (page 174)
- Upgrading the Archival Module (page 176)
- Uninstalling the Archival Module (page 177)

Prerequisites

- The procedures Creating the Core Database and Creating the Archive Database in Appendix A have been performed.
- The machine that runs the Archival Module and the machine that hosts the Core Database meet the software requirements in Chapter 2.

Installing the Archival Module

Prerequisites

- The procedures Creating the Core Database and Creating the Archive Database in Appendix A have been performed.
- The machine that runs the Archival Module and the machine that hosts the Core Database meet the software requirements in Chapter 2.

Action

To install the Archival Module:

1. Copy the directory called Archival Module onto the machine from which you will run the Archival Module.
2. If you wish, give that directory a shorter name (e.g., adam). That directory will be referred to as Archival_Module_HOME.
3. If you are in a Linux environment:
   a. Open the file "bin/silAWSArchive.sh".
   b. Assign "ADAM_HOME" the value of the variable [Archival_Module_HOME].
   c. Assign "JAVA_HOME" the correct value.
   d. Save and close the file.
4. If you are in a Windows environment:
   a. Open the file "bin/silAWSArchive.cmd".
   b. Assign "ADAM_HOME" the value of the variable [Archival_Module_HOME].

This chapter's procedures assume that your environment meets all relevant requirements in Chapter 2.

Table 13-1 describes files in the Archival Module's home directory provided by OneSpan.
c. Assign "JAVA_HOME" the correct value.

d. Save and close the file.

5. Depending on your Operating System, open the file
"bin/setPassword.sh" or "bin/setPassword.cmd", and specify
JAVA_HOME.

6. Use the password-setter utility ("bin/setPassword.cmd" or
"bin/setPassword.sh") to set and encrypt passwords for the
Production and Archive Databases.

**EXAMPLE 1:** In a Linux environment you can open a command-line
prompt, go to the "bin" sub-directory, and successively execute the
following commands (for the Production and Archive Databases,
respectively):

- setPassword ../commandline.properties
  awsng.db_source_password [password]
- setPassword ../commandline.properties
  awsng.db_target_password [password]

**EXAMPLE 2:** In a Windows environment you can open a command-
line prompt, go to the "bin" sub-directory, and successively execute
the following commands (for the Production and Archive Databases,
respectively):

- setPassword.cmd ..\commandline.properties
  awsng.db_source_password [password]
- setPassword.cmd ..\commandline.properties
  awsng.db_target_password [password]

The setPassword.sh and setPassword.cmd files use the
password specified on the command line. They overwrite the
password specified in the configuration file used by the
Archival Module (a *.properties file that, by default, is
commandline.properties).

7. Open the Archival Module’s Configuration File (by default,
[Archival_Module_HOME]/commandline.properties), and specify
its parameter values using Chapter 2 of the Archival Module User’s
Guide.

8. Open the file logging.properties, and change the line

    java.util.logging.FileHandler.pattern=adam%u.log

to

    java.util.logging.FileHandler.pattern=archive%u.log

This procedure ends here.

| **Table 13-1:** Files in the Archival Module’s home directory |
|-------------------|----------------------------------|
| **FILE**          | **DESCRIPTION**                  |
| cmdline.properties| Configuration file used by the Archival Module |
| logging.properties| Configuration file for the Archival Module’s log module |
Upgrading the Archival Module

This section tells you how to upgrade the Archival Module to OneSpan Sign 7.x from an earlier version of OneSpan Sign.

To upgrade the Archival Module to OneSpan Sign 7.x:

1. Locate the Archival Module’s home directory.
2. Make a copy of this directory in a safe location.
3. Optional: Manually delete the Archival Module’s home directory.
4. If you are using the Installer to install the Archival Module, perform the procedure in Chapter 3.
5. If you are manually installing the Archival Module, perform the procedure in the previous section.
6. Use the parameter values in the old configuration file used by the Archival Module (silAWSArchive.cfg) to specify the values in the corresponding new configuration file (a *.properties file that, by default, is the "commandline.properties" file provided by OneSpan).

This procedure ends here.

<table>
<thead>
<tr>
<th>FILE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>message.properties</td>
<td>Contains the error messages used by the Archival Module</td>
</tr>
<tr>
<td>silanisLicense.xml</td>
<td>License file that authorizes the use of the Archival Module</td>
</tr>
</tbody>
</table>

Files in the bin sub-directory

<table>
<thead>
<tr>
<th>FILE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWSArchive.jar</td>
<td>Java code for the Archival Module</td>
</tr>
<tr>
<td>PasswordSetter.jar</td>
<td>Part of the password-setter utility</td>
</tr>
<tr>
<td>setPassword.cmd</td>
<td>Part of the password-setter utility</td>
</tr>
<tr>
<td>setPassword.sh</td>
<td>Part of the password-setter utility</td>
</tr>
<tr>
<td>silAWSArchive.cmd</td>
<td>Editable command file used to launch the Archival Module in a Windows environment</td>
</tr>
<tr>
<td>silAWSArchive.sh</td>
<td>Editable command file used to launch the Archival Module in a Linux environment</td>
</tr>
</tbody>
</table>
Uninstalling the Archival Module

To uninstall the Archival Module:

1. Perform the following related steps:
   a. Make a backup copy of the Archival Module’s home directory.

   ![Warning]
   If you do not have a backup copy of this directory, you will permanently lose all its data.

   b. Delete the Archival Module’s home directory and all its contents.

2. Delete the Archival Module’s Archive Database and all its contents.

   This procedure ends here.
CHAPTER 14: Working with the e-Vault Manager

The e-Vault Manager is a companion server product to OneSpan Sign that can create, control, and transfer ownership of the financial instruments called Transferable Records (e.g., e-Notes in a mortgage process).

Prerequisites

- Your environment meets all relevant requirements in Chapter 2.
- You are licensed to use the e-Vault Manager™.
- The e-Vault Manager™ has been installed (see the e-Vault Manager™ Deployment Guide).

⚠️ If you are using JBoss/Wildfly and are deploying both OneSpan Sign and the e-Vault Manager, you must install OneSpan Sign and the e-Vault Manager on two different JBoss/Wildfly servers.

- The procedure in Appendix C (Configuring Cryptographic Libraries) has been performed.

Action

To configure OneSpan Sign to work with the e-Vault Manager:

- Configure the e-Vault Manager using the Platform Administrator’s Guide (see Table C-8 of that guide).

This procedure ends here.
### PART V: APPLICATION COMPONENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>15: <em>Deploying Application Software</em></td>
<td>180</td>
</tr>
</tbody>
</table>
CHAPTER 15: Deploying Application Software

The Application is essentially the front end that before OneSpan Sign 6.0 was available only via a SaaS deployment. Users, administrators, and integrators can interact with the Application using its GUI, its API, its SDKs, or its connectors.

To deploy the Application's components, see the following sections:

• Deploying Multiple Application Components (page 180)
• Deploying the Transaction Status API (page 195)
• Deploying the OneSpan Sign Application Frontend (page 195)
• Deploying the Trusted Service Provider (page 217)
• Deploying the OneSpan Sign Document Converter (page 228)
• Installing Connectors for the Application (page 230)
• Installing the Personal Certificate Client (page 233)

Deploying Multiple Application Components

This section describes how to deploy most OneSpan Sign Application components. Specifically, it describes how to install and uninstall the following:

1. OneSpan Sign Application Backend
2. OneSpan Sign BackOffice
3. BackOffice Support
4. OneSpan Sign Authentication
5. OneSpan Sign Scheduler
6. OneSpan Sign e-Notary Validation
7. OneSpan Sign Equifax
8. OneSpan Sign Single Sign-On
9. OneSpan Sign Mobile SDK
10. OneSpan Sign URL Mapper

Items 1-5 in the above list are required by the Application, in the sense that the Application will not work properly unless they're installed. Items 6-10 are optional.

The remainder of this section discusses:

• Installing Multiple Application Components (page 181)
• Uninstalling Multiple Application Components (page 194)
Installing Multiple Application Components

This section describes how to manually install multiple Application components. Chapter 3 describes an alternate, automated way of doing so if you are using WebSphere, JBoss, or Wildfly.

Prerequisites

- All required Platform software has been installed (see Part III).
- Your environment meets all relevant requirements in Chapter 2.
- The procedures Creating the Application Database (page 288) and Preparing an App Server Environment (page 312) have been performed.
- Ensure that the JAVA character encoding on the Application Server is UTF-8 (file.encoding=UTF-8).
- The Transaction Status API has been installed (see the Transaction Status API Deployment Guide provided by OneSpan).
- If you want to change the number of permitted concurrent sessions for a particular jms-listener, add the appropriate one of the following system variables, and specify the desired number after the equal sign:
  - concurrency_asyncBusinessEventCallbackConsumer=1
  - concurrency_asyncBusinessEventNotificationConsumer=1
  - concurrency_asynchronousMessageConsumerForEmail=5
  - concurrency_asynchronousMessageConsumerForSms=5
  - concurrency_businessEventQueueConsumer=1
  - concurrency_connectorsFrameworkCallbackEventMessageListener=1
  - concurrency_evidenceSummaryJmsListener=1
  - concurrency_expireMessageListener=1
  - concurrency_packageEventMessageListener=5
  - concurrency_packageRemindersMessageListener=1
  - concurrency_statisticMessageListener=1

The number after each equal sign above is the default number of permitted concurrent sessions for the jms-listener.

- If you are upgrading OneSpan Sign from version 6.0, 6.0.1, or 6.0.2 to version 6.1 or higher, the following Java runtime argument has been set (to ensure the backwards compatibility of encryption algorithms):
  -Dspring.profiles.active=production,external,encryption-compatibility
- If you are using WebSphere, the following JVM parameter has been set:
  org.apache.el.parser.SKIP_IDENTIFIER_CHECK=true
  This setting will enable you to install OneSpan Sign BackOffice.
- If your Application Server user does not have a defined HOME directory, we recommend that you define the system variable pdfbox.fontcache to point to a directory where that user has full privileges (e.g., on Linux, that directory could be /tmp/).
• If your Database Server is Microsoft SQL Server, you have configured the following flags:

```
ALTER DATABASE <DB_name> SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE <DB_name> SET READ_COMMITTED_SNAPSHOT ON
```

⚠️ All names assigned in the next bullet must match exactly those written below.

• If you are using WebLogic:
  a. Create a JMS server called eSLJmsServer.
  b. Create a JMS module called eSLJmsModule.
  c. From the eSLJmsModule, create a sub-deployment called eSLSubdeployment, selecting eSLJmsServer as its target.
  d. Add the following system variables:

```
• statisticsQueue=eSLJmsServer/eSLJmsModule!statisticsQueue
• expirePackageQueue=eSLJmsServer/eSLJmsModule!expirePackageQueue
• packageCallbackQueue=eSLJmsServer/eSLJmsModule!packageCallbackQueue
• connectorsFrameworkCallbackQueue=eSLJmsServer/eSLJmsModule!connectorsFrameworkCallbackQueue
• salesforcePackageEventQueue=eSLJmsServer/eSLJmsModule!salesforcePackageEventQueue
• salesforceUploadCompletedQueue=eSLJmsServer/eSLJmsModule!salesforceUploadCompletedQueue
• evidenceSummaryQueue=eSLJmsServer/eSLJmsModule!evidenceSummaryQueue
• packageRemindersQueue=eSLJmsServer/eSLJmsModule!packageRemindersQueue
• emailQueue=eSLJmsServer/eSLJmsModule!emailQueue
• smsQueue=eSLJmsServer/eSLJmsModule!smsQueue
• asyncBusinessQueue=eSLJmsServer/eSLJmsModule!asyncBusinessQueue
• esl.app.jta.platform=org.hibernate.service.jta.platform.internal.WeblogicJtaPlatform
```

The packageCallbackQueue is for the Push Notification Service (for more on that, see the Connectors Guide).

• If you are using WebSphere, add the following system variables:

```
• esl.app.jta.platform=org.hibernate.service.jta.platform.internal.WebSphereExtendedJtaPlatform
```
Installing Multiple Application Components

- `esl.app.hibernate.transaction.factory_class=org.hibernate.engine.transaction.internal.jta.JtaTransactionFactory`

- If you are using WebSphere and an Oracle database, add the following system variable:
  - `esl.app.hibernate.dialect=org.hibernate.dialect.Oracle10gDialect`

**Action**

To install most *OneSpan Sign Application* components:

1. Create four Data Sources for the Application Database using: (i) the parameters in Table 15-1; (ii) the appropriate procedure from *Creating a JDBC Provider* (page 316). Ensure that *Container Managed Persistence* (CMP) is *not* selected.

   The user account for the Data Source should have read/write access to the *OneSpan Sign Application* Database.

2. If you are using WebSphere SIB default messaging:
   a. Create a bus for the *OneSpan Sign Application Backend*, and assign as its "members" the servers or clusters on which the *OneSpan Sign Application Backend* will run. If you wish, configure bus security.

   *Step 3 and Step 4* use the bus created in Step 2a.

   b. Create the following destinations within the bus:

   - `statisticsQueue`
   - `expirePackageQueue`
   - `packageCallbackQueue`
   - `connectorsFrameworkCallbackQueue`
   - `salesforcePackageEventQueue`
   - `salesforceUploadCompletedQueue`
   - `evidenceSummaryQueue`
   - `packageRemindersQueue`
   - `emailQueue`
   - `smsQueue`
   - `asyncBusinessQueue`
3. To continue the configuration of JMS resources:
   
a. Create a **Connection Factory** with JNDI name "JmsXA", setting the
      Client ID to "eSL-JMS-client".

      If you are using WebSphere SIB default
      messaging, the **Client ID** is in the **Advanced**
      section. If you are using WebSphere MQ
      messaging, the equivalent **Client identifier** is
      in the **Durable Subscription** section.

   b. Create a **Connection Factory** with JNDI name
      "jms/ConnectionFactory", setting the Client ID to "eSL-JMS-
      client2".

c. Create the following persistent JMS queues:
   
   • statisticsQueue (use the parameters in Table 15-2)
   • expirePackageQueue (use the parameters in Table 15-3)
   • packageCallbackQueue (use the parameters in Table 15-4)
   • connectorsFrameworkCallbackQueue (use the parameters in
     Table 15-5)
   • salesforcePackageEventQueue (use the parameters in Table 15-6)
   • salesforceUploadCompletedQueue (use the parameters in Table 15-
     7)
   • evidenceSummaryQueue (use the parameters in Table 15-8)
   • packageRemindersQueue (use the parameters in Table 15-9)
   • emailQueue (use the parameters in Table 15-10)
   • smsQueue (use the parameters in Table 15-11)

      If you are using WebLogic, for all the above
      queues you must set: (1) **Redelivery Limit = 3**; (2) **Redelivery Delay Override = 5000** (= 5 seconds).

   • asyncBusinessQueue (use the parameters in Table 15-12)

      If you are using WebSphere, for each of the above
      queues: (1) from the WebSphere Administration
      Console, go to Resources > JMS > Queues >
      Queue name > Advanced properties; (2) select
      the checkbox **Append RFH version 2 headers to messages sent to this destination Message**.

4. If you are using WebLogic, assign the Connection Factories and the
   JMS queues created in the preceding step to the sub-deployment
   eSLSubdeployment.

5. If you’re using WebSphere, complete the configuration of JMS
   resources by doing the following:
   
a. Create the following activation specification for the Statistics
      Queue:

      • **Name**: ActivationSpecStatistics
b. Create the following activation specification for the Expire Package Queue:
   • Name: ActivationSpecExpirePackage
   • JNDI name: jms/ActivationSpecExpirePackage
   • Destination JNDI name: jms/queue/expirePackageQueue

c. Create the following activation specification for the Package Callback Queue:
   • Name: ActivationSpecPackageCallback
   • JNDI name: jms/ActivationSpecPackageCallback
   • Destination JNDI name: jms/queue/packageCallbackQueue

d. Create the following activation specification for the Connectors Framework Callback Queue:
   • Name: ActivationSpecConnectorsFrameworkCallback
   • JNDI name: jms/ActivationSpecConnectorsFrameworkCallback
   • Destination JNDI name: jms/queue/connectorsFrameworkCallbackQueue

e. Create the following activation specification for the Salesforce Package Event Queue:
   • Name: ActivationSpecSalesforcePackageEvent
   • JNDI name: jms/ActivationSpecSalesforcePackageEvent
   • Destination JNDI name: jms/queue/salesforcePackageEventQueue

f. Create the following activation specification for the Salesforce Upload Completed Queue:
   • Name: ActivationSpecSalesforceUploadCompleted
   • JNDI name: jms/ActivationSpecSalesforceUploadCompleted
   • Destination JNDI name: jms/queue/salesforceUploadCompletedQueue

g. Create the following activation specification for the Evidence Summary Queue:
   • Name: ActivationSpecEvidenceSummary
   • JNDI name: jms/ActivationSpecEvidenceSummary
   • Destination JNDI name: jms/queue/evidenceSummaryQueue

h. Create the following activation specification for the Package Reminders Queue:
   • Name: ActivationSpecPackageReminders
   • JNDI name: jms/ActivationSpecPackageReminders
   • Destination JNDI name: jms/queue/packageRemindersQueue
i. Create the following activation specification for the Email Queue:
   • Name: ActivationSpecEmail
   • JNDI name: jms/ActivationSpecEmail
   • Destination JNDI name: jms/queue/emailQueue

j. Create the following activation specification for the SMS Queue:
   • Name: ActivationSpecSMS
   • JNDI name: jms/ActivationSpecSMS
   • Destination JNDI name: jms/queue/smsQueue

k. Create the following activation specification for the asyncBusiness Queue:
   • Name: ActivationSpecAsyncBusinessQueue
   • JNDI name: jms/ActivationSpecAsyncBusinessQueue
   • Destination JNDI name: jms/queue/asyncBusinessQueue

6. Do one of the following:
   • If you are using an Oracle database, add the following system variable:
     esl.app.hibernate.dialect=org.hibernate.dialect.Oracle10gDialect
   • If you are using an IBM DB2 database, add the following system variable:
     esl.app.hibernate.dialect=org.hibernate.dialect.DB2Dialect
   • If you are using a Microsoft SQL Server database whose version number is lower than 2012, add the following system variable:
     esl.app.hibernate.dialect=org.hibernate.dialect.SQLServerDialect
   • If you are using a Microsoft SQL Server database whose version number is 2012 or higher, add the following system variable:
     esl.app.hibernate.dialect=com.silanis.esl.hibernate.dialect.Cust omSqlServerDialect
   • If you are using a MySQL database, add the following system variable:
     esl.app.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect

7. Inside the relevant Application Servers depicted in Figure 2-4, deploy the following *.ear files provided by OneSpan:
   • esl.ear (for the OneSpan Sign Application Backend)
     During the manual deployment of esl.ear, the Deployment Manager will ask for the Reviewer mapping. Click Browse, and select jboss/jdbc/ConnectionPool (the Data Source created in Step 1).
     When you deploy esl.ear and backoffice.ear, you must configure their class loader so classes are loaded with parent class loader first.
   • backoffice.ear (for OneSpan Sign BackOffice)
• esl-auth.ear (for OneSpan Sign Authentication)
• esl-internal.ear (for BackOffice Support)
• One of the following (for the OneSpan Sign Scheduler):
  • WebSphere: batch-was.ear
  • Any other supported App Server: batch.ear

Even if you are in a clustered environment, there should be only one instance of the OneSpan Sign Scheduler and only one instance of OneSpan Sign BackOffice.

8. Optional: Inside the relevant Application Server depicted in Figure 2-4, deploy any of the following OneSpan-provided archive files you want:
• enotary-validation-service.war (for OneSpan Sign e-Notary Validation)
• equifax-service.war (for OneSpan Sign Equifax)
• MobileApi.war (for the OneSpan Sign Mobile SDK)

During the manual deployment of MobileApi.war, the Deployment Manager will ask for the Context Root. Input: /aws/rest/mobile

• url-mapper.ear (for the OneSpan Sign URL Mapper)
• sso.ear (for OneSpan Sign Single Sign-On)

OneSpan Sign Single Sign-On will not work until you have configured SAML. For details, see Chapter 18.

9. Restart all Application Servers (see Figure 2-4).

After restarting, many errors will appear in your log files. These should be harmless, appearing only because you have not yet configured the system.

10. Configure the Application by performing the procedure in Chapter 18.

If you don’t configure Table 18-1’s parameters in Step 10, log-file errors will ask you to do so when you try to restart the Application Servers in Step 11.
11. Clear your log files. Then restart all Application Servers (see Figure 2-4).

If after installing the OneSpan Sign Application Backend, you experience Reporting timeout exceptions, assign a suitable value to the JVM parameter search.wait.timeout.seconds.

The product supports using an independent time-stamping service to:
(1) validate that the certificate used for signing has not expired;
(2) embed that information in the signature’s Audit Trail. This capability is not enabled by default. It requires a specific certificate type, and a subscription to a time-stamping service. To learn more, please contact OneSpan.

This procedure ends here.

Table 15-1: OneSpan Application & URL Mapper Data Source parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the OneSpan Sign Application Data Source</td>
<td>Any character string</td>
</tr>
<tr>
<td>WebSphere calls this parameter the Data source name.</td>
<td></td>
</tr>
<tr>
<td>JNDI Name associated with the first (of two) OneSpan Sign Application Data Sources</td>
<td>This name must be: jboss/jdbc/ConnectionPool</td>
</tr>
<tr>
<td>Note: This Data Source must support global transactions with a Two-Phase Commit.</td>
<td></td>
</tr>
<tr>
<td>JNDI Name associated with the second (of two) OneSpan Sign Application Data Sources</td>
<td>This name must be: jboss/jdbc/reportingDS</td>
</tr>
<tr>
<td>Note: This Data Source must support global transactions with a Two-Phase Commit.</td>
<td></td>
</tr>
<tr>
<td>JNDI Name associated with the OneSpan URL Mapper Data Source</td>
<td>This name must be: jboss/jdbc/UrlMapper</td>
</tr>
<tr>
<td>JNDI Name associated with the OneSpan Sign BackOffice Data Source</td>
<td>This name must be: jboss/jdbc/EslBackoffice</td>
</tr>
<tr>
<td>Note: This Data Source must support global transactions with a Two-Phase Commit.</td>
<td></td>
</tr>
<tr>
<td>Transaction Isolation Level</td>
<td>READ_COMMITTED</td>
</tr>
<tr>
<td>For example, for WebSphere add a datasource custom property called webSphereDefaultIsolationLevel, and set its value to 2.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 15-2: Statistics Queue parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Java Context</td>
<td>The value of <code>&lt;use-java-context&gt;</code> must be false.</td>
</tr>
<tr>
<td>Note: This parameter is needed only if you are using JBoss or Wildfly.</td>
<td></td>
</tr>
<tr>
<td>The Driver Class</td>
<td>Note: This parameter is needed only if you are using JBoss or Wildfly.</td>
</tr>
<tr>
<td>The value of <code>&lt;driver-class&gt;</code> must be as follows for the three possible databases:</td>
<td></td>
</tr>
<tr>
<td>Driver Class</td>
<td></td>
</tr>
<tr>
<td>• Oracle — <code>oracle.jdbc.driver.OracleDriver</code></td>
<td></td>
</tr>
<tr>
<td>• DB2 — <code>com.ibm.db2.jcc.DB2Driver</code></td>
<td></td>
</tr>
<tr>
<td>• SQLServer — <code>com.microsoft.sqlserver.jdbc.SQLServerDriver</code></td>
<td></td>
</tr>
<tr>
<td>• MySQL — <code>com.mysql.jdbc.Driver</code></td>
<td></td>
</tr>
<tr>
<td>Definition of the JBoss/Wildfly driver referenced by the Audit Service Data Source</td>
<td>Possible values:</td>
</tr>
<tr>
<td>• <code>oracle.ojdbc</code> (Oracle)</td>
<td></td>
</tr>
<tr>
<td>• <code>ibm.db2</code> (DB2)</td>
<td></td>
</tr>
<tr>
<td>• <code>com.microsoft</code> (MS SQL Server)</td>
<td></td>
</tr>
<tr>
<td>• <code>mysql</code> (MySQL)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Statistics Queue</td>
<td>statisticsQueue</td>
</tr>
<tr>
<td>WebSphere calls this parameter the <em>Base queue name</em>.</td>
<td></td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
<td>statisticsQueue</td>
</tr>
<tr>
<td>JNDI name associated with the Statistics Queue</td>
<td>The JNDI name must contain the queue name as follows. If the queue name is &quot;statisticsQueue&quot;, the JNDI name must be &quot;jms/queue/statisticsQueue&quot;.</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
<td>Persistent</td>
</tr>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
<td>2</td>
</tr>
<tr>
<td>WebSphere MQ and WebSphere call this parameter the <em>Backout threshold</em>.</td>
<td></td>
</tr>
</tbody>
</table>
**Table 15-3: Expire Package Queue parameters**

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Expire Package Queue</td>
<td>expirePackageQueue</td>
</tr>
<tr>
<td>WebSphere calls this parameter the <em>Base queue name.</em></td>
<td>expirePackageQueue</td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
<td>expirePackageQueue</td>
</tr>
<tr>
<td>JNDI name associated with the Expire Package Queue</td>
<td>The JNDI name must contain the queue name as follows. If the queue name is &quot;expirePackageQueue&quot;, the JNDI name must be &quot;jms/queue/expirePackageQueue&quot;.</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
<td>Persistent</td>
</tr>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
<td>2</td>
</tr>
<tr>
<td>WebSphere MQ and WebSphere call this parameter the <em>Backout threshold.</em></td>
<td>2</td>
</tr>
</tbody>
</table>

**Table 15-4: Package Callback Queue parameters**

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Package Callback Queue</td>
<td>packageCallbackQueue</td>
</tr>
<tr>
<td>WebSphere calls this parameter the <em>Base queue name.</em></td>
<td>packageCallbackQueue</td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
<td>packageCallbackQueue</td>
</tr>
<tr>
<td>JNDI name associated with the Package Callback Queue</td>
<td>The JNDI name must contain the queue name as follows. If the queue name is &quot;packageCallbackQueue&quot;, the JNDI name must be &quot;jms/queue/packageCallbackQueue&quot;.</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
<td>Persistent</td>
</tr>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
<td>2</td>
</tr>
<tr>
<td>WebSphere MQ and WebSphere call this parameter the <em>Backout threshold.</em></td>
<td>2</td>
</tr>
</tbody>
</table>

**Table 15-5: Connector Framework Callback Queue parameters**

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Connector Framework Callback Queue</td>
<td>connectorsFrameworkCallbackQueue</td>
</tr>
<tr>
<td>WebSphere calls this parameter the <em>Base queue name.</em></td>
<td>connectorsFrameworkCallbackQueue</td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
<td>connectorsFrameworkCallbackQueue</td>
</tr>
</tbody>
</table>
### Table 15-6: Salesforce Package Event Queue parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>JNDI name associated with the Connector Framework Callback Queue</td>
<td>The JNDI name must contain the queue name as follows. If the queue name is &quot;connectorsFrameworkCallbackQueue&quot;, the JNDI name must be &quot;jms/queue/connectorsFrameworkCallbackQueue&quot;.</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
<td>Persistent</td>
</tr>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
<td>2</td>
</tr>
<tr>
<td>WebSphere MQ and WebSphere call this parameter the Backout threshold.</td>
<td></td>
</tr>
</tbody>
</table>

### Table 15-7: Salesforce Upload Completed Queue parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Salesforce Package Event Queue</td>
<td>salesforcePackageEventQueue</td>
</tr>
<tr>
<td>WebSphere calls this parameter the Base queue name.</td>
<td>salesforcePackageEventQueue</td>
</tr>
<tr>
<td>JNDI name associated with the Salesforce Package Event Queue</td>
<td>The JNDI name must contain the queue name as follows. If the queue name is &quot;salesforcePackageEventQueue&quot;, the JNDI name must be &quot;jms/queue/salesforcePackageEventQueue&quot;.</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
<td>Persistent</td>
</tr>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
<td>2</td>
</tr>
<tr>
<td>WebSphere MQ and WebSphere call this parameter the Backout threshold.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 15-8: Evidence Summary Queue parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Evidence Summary Queue</td>
<td>evidenceSummaryQueue</td>
</tr>
<tr>
<td>WebSphere calls this parameter the Base queue name.</td>
<td>evidenceSummaryQueue</td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
<td>evidenceSummaryQueue</td>
</tr>
<tr>
<td>JNDI name associated with the Evidence Summary Queue</td>
<td>The JNDI name must contain the queue name as follows. If the queue name is &quot;evidenceSummaryQueue&quot;, the JNDI name must be &quot;jms/queue/evidenceSummaryQueue&quot;.</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
<td>Persistent</td>
</tr>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
<td>2</td>
</tr>
<tr>
<td>WebSphere MQ and WebSphere call this parameter the Backout threshold.</td>
<td></td>
</tr>
</tbody>
</table>

### Table 15-9: Package Reminders Queue parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Package Reminders Queue</td>
<td>packageRemindersQueue</td>
</tr>
<tr>
<td>WebSphere calls this parameter the Base queue name.</td>
<td>packageRemindersQueue</td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
<td>packageRemindersQueue</td>
</tr>
<tr>
<td>JNDI name associated with the Package Reminders Queue</td>
<td>The JNDI name must contain the queue name as follows. If the queue name is &quot;packageRemindersQueue&quot;, the JNDI name must be &quot;jms/queue/packageRemindersQueue&quot;.</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
<td>Persistent</td>
</tr>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
<td>2</td>
</tr>
<tr>
<td>WebSphere MQ and WebSphere call this parameter the Backout threshold.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 15-10: Email Queue parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Email Queue</td>
<td>emailQueue</td>
</tr>
<tr>
<td>WebSphere calls this parameter the <em>Base queue name</em>.</td>
<td>emailQueue</td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
<td>emailQueue</td>
</tr>
<tr>
<td>JNDI name associated with the Email Queue</td>
<td>The JNDI name must contain the queue name as follows. If the queue name is &quot;emailQueue&quot;, the JNDI name must be &quot;jms/queue/emailQueue&quot;.</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
<td>Persistent</td>
</tr>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
<td>2</td>
</tr>
<tr>
<td>WebSphere MQ and WebSphere call this parameter the <em>Backout threshold</em>.</td>
<td></td>
</tr>
</tbody>
</table>

### Table 15-11: SMS Queue parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the SMS Queue</td>
<td>smsQueue</td>
</tr>
<tr>
<td>WebSphere calls this parameter the <em>Base queue name</em>.</td>
<td>smsQueue</td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
<td>smsQueue</td>
</tr>
<tr>
<td>JNDI name associated with the SMS Queue</td>
<td>The JNDI name must contain the queue name as follows. If the queue name is &quot;smsQueue&quot;, the JNDI name must be &quot;jms/queue/smsQueue&quot;.</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
<td>Persistent</td>
</tr>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
<td>2</td>
</tr>
<tr>
<td>WebSphere MQ and WebSphere call this parameter the <em>Backout threshold</em>.</td>
<td></td>
</tr>
</tbody>
</table>

### Table 15-12: Async Business Queue parameters

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the Async Business Queue</td>
<td>asyncBusinessQueue</td>
</tr>
<tr>
<td>WebSphere calls this parameter the <em>Base queue name</em>.</td>
<td>asyncBusinessQueue</td>
</tr>
<tr>
<td>WebSphere MQ Queue</td>
<td>asyncBusinessQueue</td>
</tr>
</tbody>
</table>
Uninstalling Multiple Application Components

To uninstall most OneSpan Sign Application components:

1. Inside the Application Server, uninstall any of the following optional applications that were installed during the procedure Installing Multiple Application Components (page 181):
   - enotary-validation-service.war
   - equifax-service.war
   - sso.ear
   - MobileApi.war

2. Inside the Application Server, uninstall the following applications:
   - esl.ear
   - esl-auth.ear
   - backoffice.ear
   - batch.ear or batch-was.ear

3. If you’re using WebSphere, delete the activation specifications created during the procedure Installing Multiple Application Components (page 181).

4. Delete the JMS queues created during the procedure Installing Multiple Application Components (page 181).

5. Delete the Connection Factories created during the procedure Installing Multiple Application Components (page 181).

6. If you used WebSphere SIB default messaging:
   a. Remove the bus members and destinations created during the procedure Installing Multiple Application Components (page 181).
   b. Delete the bus created during the procedure Installing Multiple Application Components (page 181).

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>JNDI name associated with the Async Business Queue</td>
<td>The JNDI name must contain the queue name as follows. If the queue name is &quot;asyncBusinessQueue&quot;, the JNDI name must be &quot;jms/queue/asyncBusinessQueue&quot;.</td>
</tr>
<tr>
<td>Queue persistence delivery mode</td>
<td>Persistent</td>
</tr>
<tr>
<td>Maximum number of retries when a queue fails to consume a message</td>
<td>2</td>
</tr>
<tr>
<td>WebSphere MQ and WebSphere call this parameter the Backout threshold.</td>
<td></td>
</tr>
</tbody>
</table>

Uninstalling Multiple Application Components

This section describes how to manually uninstall eight Application components. Chapter 3 describes an alternate, automated way of doing so if you are using WebSphere, JBoss, or Wildfly.

To uninstall most OneSpan Sign Application components:

1. Inside the Application Server, uninstall any of the following optional applications that were installed during the procedure Installing Multiple Application Components (page 181):
   - enotary-validation-service.war
   - equifax-service.war
   - sso.ear
   - MobileApi.war

2. Inside the Application Server, uninstall the following applications:
   - esl.ear
   - esl-auth.ear
   - backoffice.ear
   - batch.ear or batch-was.ear

3. If you’re using WebSphere, delete the activation specifications created during the procedure Installing Multiple Application Components (page 181).

4. Delete the JMS queues created during the procedure Installing Multiple Application Components (page 181).

5. Delete the Connection Factories created during the procedure Installing Multiple Application Components (page 181).

6. If you used WebSphere SIB default messaging:
   a. Remove the bus members and destinations created during the procedure Installing Multiple Application Components (page 181).
   b. Delete the bus created during the procedure Installing Multiple Application Components (page 181).
7. If you used WebSphere MQ messaging, delete the queues created on the MQ server in Step 7 of Preparing an Environment for WebSphere (page 312).

8. Delete the Data Source for the Application Database.

9. Delete the JDBC Provider for the Application Database (provided it’s not being used by any other application).

10. If you are using WebSphere or JBoss/Wildfly, and if you copied any JDBC driver files for the Application Database during the procedure Installing Multiple Application Components (page 181), delete them (provided they’re not being used by any other application).

11. If you are using WebLogic, if you were not using an Oracle database during the procedure Installing Multiple Application Components (page 181), and if you copied any JDBC driver files for the Application Database at that time, delete them (provided they’re not being used by any other application).

12. Perform the following related steps:
   a. Make a backup copy of the Application Database.

   ![Warning]
   If you do not have a backup copy of this database, you will permanently lose all its data.

   b. Delete the Application Database and all its contents.

13. Restart the Application Server.

   This procedure ends here.

### Deploying the Transaction Status API

The Transaction Status API is an API that enables integrators to build an external GUI to monitor and create reports on OneSpan Sign transactions.

To install or uninstall the Transaction Status API, follow the instructions in the Transaction Status API Deployment Guide provided by OneSpan.

![Note]
The Transaction Status API is required by e-Witness and by the Application.

### Deploying the OneSpan Sign Application Frontend

The OneSpan Sign Application Frontend is the Application’s User Experience (UX). OneSpan Sign actually offers two UXs, whose deployment is described in the
following sections:

- Deploying the Classic User Experience (page 196)
- Deploying the New User Experience (page 206)

**Deploying the Classic User Experience**

The *Classic User Experience* is the Application UX for signers, senders and administrators that became available in *OneSpan Sign 6.0*. That UX is still available. Its use is described in the *Classic Application User’s Guide*.

This section describes:

- Installing the Classic User Experience (page 196)
- Uninstalling the Classic User Experience (page 206)

**Installing the Classic User Experience**

The following procedure creates a user called *eSignLive*. The entire *OneSpan Sign Application Frontend* is installed under that user (i.e., all that component’s files will be owned by that user).

**Prerequisites**

- The procedure *Installing Multiple Application Components* (page 181) has been performed.
- Your environment meets all relevant requirements in *Chapter 2*.
- You have opened a port in your firewall for use by the *OneSpan Sign Application Frontend* (e.g., port 443), and have forwarded that port to port 80 of the *OneSpan Sign Application Frontend*. That port in your firewall must be listening using an SSL certificate. You have also opened: (1) port 85 for *mobile* e-signature processes; (2) internal port 1443 for e-Witness.
- Ensure that Apache is permitted to make outbound connections to the *OneSpan Sign Application Backend*.
- The "SELinux" policy is set to *Permissive* or *Disabled*.
- Because the *OneSpan Sign Application Frontend* uses encrypted cookies to store user information, the traffic between the browser and the server (Apache, or an external Load Balancer) must be HTTPS.
- The superuser (username = root) is available to perform the installation.
- If you are using RHEL 6 or CentOS 6 with *JBoss Core Services*, the following packages have been installed on your system:
  - jbcs-httpd24-httpd-zip
  - apr-jw3
  - apr-jws3-devel
  - apr-util-jw3
  - apr-util-jws3-devel
• If you have an existing installation of the *Classic user Experience*:
  a. Shut down the Web Application Server (s) running Apache 2.x.

  The previous step requires downtime unless your environment has multiple instances of the *Classic User Experience*, and you choose to leave at least one of them running.

  b. Uninstall your existing *Classic User Experience* using the procedure *Uninstalling the Classic User Experience* (page 206).

  c. Remove the following block of code (located in
     `/etc/httpd/conf/httpd.conf`):

     ```
     LoadModule passenger_module /usr/local/rvm/gems/ruby-2.5.8/gems/passenger-5.0.16/buildout/apache2/mod_passenger.so
     <IfModule mod_passenger.c>
     PassengerRoot /usr/local/rvm/gems/ruby-2.5.8
     /gems/passenger-5.0.16
     PassengerDefaultRuby /usr/local/rvm/gems/ruby-2.5.8
     /wrappers/ruby
     </IfModule>
     ```

  • If you are using Ruby only for the *Classic User Experience*, uninstall it by running the following commands:
    a. `yum remove ruby`
    b. `rm -rf /usr/lib64/ruby/`

  • The RHEL/CentOS repositories must be accessible from the *OneSpan Sign Application Frontend* machine.

    *We strongly* recommend that you install the *OneSpan Sign Application Frontend* on its own dedicated machine with a standard RHEL/CentOS 6 or 7 installation.

  The installer of the *OneSpan Sign Application Frontend* does not currently support automatically configuring 1-way or 2-way SSL connections.

**Action**

**To install and configure the *Classic User Experience***:

1. Do one of the following:

   • If you are using RHEL 6 or CentOS 6, locate the file "release-<version #>-frontend-el6.tar.gz" provided by *OneSpan*.
   • If you are using RHEL 7 or CentOS 7, locate the file "release-
Deploying the Classic User Experience

<version #>-frontend-el7.tar.gz" provided by OneSpan.

In the above step and elsewhere in this procedure, <version #> is a placeholder for an internal version number (e.g., 16.3.10).

2. Copy that file into the OneSpan Sign Application Frontend machine.
3. Log on to that machine, and go to the directory where you copied that file.
4. Do one of the following:
   - If you are using RHEL 6 or CentOS 6 without JBoss Core Services, run the following commands:
     mkdir release
     tar -xvf release-<version #>-frontend-el6.tar.gz -C release
     cd release
     ./install.sh
   - If you are using RHEL 6 or CentOS 6 with JBoss Core Services, run the following commands:
     mkdir release
     tar -xvf release-<version #>-frontend-el6.tar.gz -C release
     cd release
     ./install.sh <Directory in JBoss Core Services of Apache’s configuration file ‘httpd.conf’>
5. If you are using RHEL 7 or CentOS 7, run the following commands:
   mkdir release
   tar -xvf release-<version #>-frontend-el7.tar.gz -C release
   cd release
   ./install.sh

The preceding steps will install the Classic User Experience in the directory /opt/eSLFrontEnd/.

6. Run the following command:
   ./configure.sh http(s): //<URL_1>://<port_1>
   https://<URL_2>://<port_2> http(s): //<URL_3>://<port_3>
   http(s): //<URL_4>://85

Here:
   - <URL_1> denotes the URL of the server where the OneSpan Sign Application Backend is installed, and <port_1> denotes the port it uses.
   - <URL_2> denotes the URL of the server where the OneSpan Sign Platform is installed, and <port_2> denotes the port it uses. This
URL must be SSL-enabled.

- `<URL_3>` denotes the URL of the public-facing Domain Name of your organization, and `<port_3>` denotes the port it uses (e.g., https://apps.eSignLive.com:443). These arguments are used to communicate with the OneSpan Sign Application Frontend.

- `<URL_4>` denotes the URL of the public-facing Domain Name of your organization. This argument and port 85 are used to communicate with an e-signature process on a mobile device.

If `<port_3>` is different from 443 or 80, you must configure Apache or a Load Balancer to redirect the `<URL_3>` Domain to http(s): //<URL_3>::<port_3>.

If you have trouble starting your HTTPD server, please ensure that all of the above Prerequisites are satisfied.

7. Optional: In the directory "/opt/eSLFrontEnd/config/vhosts/", edit the file "localhost.80.vhost.conf" by appropriately specifying the parameters in Table 15-13.

8. To verify the installation:
   a. As an Administrator, verify that version 2.5.8 of Ruby has been installed on your machine (e.g., use the command: ruby -v).
   b. Navigate to the following URL:

   http(s)://<URL_2>::<port_2>

If at this point you want to change the ports for the Classic Front End, do as follows: (1) in the file eSLFrontEnd.vhost.conf, change `<VirtualHost *:80>` to `<VirtualHost *:NewPort#>`; (2) in the file eSLEvidence.vhost.conf, change `<VirtualHost *:1443>` to `<VirtualHost *:NewPort#>`; (3) in the file eSLMSC.vhost.conf, change Listen 85 to Listen NewPort#, change NameVirtualHost *:85 to NameVirtualHost *:NewPort#, and change `<VirtualHost *:85>` to `<VirtualHost *:NewPort#>`.

This procedure ends here.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>CACHE_BUSTER</td>
<td>Integer</td>
<td>Number applied to URLs, which enables &quot;cache busting&quot;</td>
<td>Default: &lt;random number&gt;</td>
</tr>
<tr>
<td>ESL_SANDBOX</td>
<td>boolean</td>
<td>Flag that determines if the environment is OneSpan Sign Sandbox</td>
<td>Possible values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• false (default)</td>
</tr>
<tr>
<td>PARAMETER</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
<td>VALUES</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ESL_ENABLE_RECAPTCHA</td>
<td>boolean</td>
<td>Flag that determines if the reCAPTCHA plugin will be used</td>
<td>Possible values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• false (default)</td>
</tr>
<tr>
<td>RECAPTCHA_SITE_KEY</td>
<td>string</td>
<td>Public site key of the reCAPTCHA application</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This parameter requires: (1) ESL_ENABLE_RECAPTCHA = true.</td>
<td>For help with this parameter, consult this FAQ.</td>
</tr>
<tr>
<td>ESL_ENABLE_GOOGLE_ANALYTICS</td>
<td>boolean</td>
<td>Flag that determines if Google Analytics will be enabled</td>
<td>Possible values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• false (default)</td>
</tr>
<tr>
<td>GOOGLE_ANALYTICS_UA</td>
<td>string</td>
<td>User account for Google Analytics</td>
<td>Default: UA-2950310-12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>URL of the <em>Sign up</em> link accessible from the Login page of the <em>Classic User Experience</em> in Sandbox mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This parameter requires: (1) ESL_FOOTER_ON_LOGIN=true; (2) disabling of the <em>Disable Footer</em> account feature (see the <em>Application Administrator’s Guide</em>).</td>
<td>No default</td>
</tr>
</tbody>
</table>
### Deploying the Classic User Experience

**ESL_SIGNUP_LINK**  
String  
URL of the *Sign up* link accessible from the Login page of the *Classic User Experience* in Production mode.  
**Note:** This parameter requires: (1) ESL_FOOTER_ON_LOGIN=true; (2) disabling of the *Disable Footer* account feature (see the Application Administrator’s Guide).

**REMOVE_SIGN_UP**  
Boolean  
Flag that determines if the **ESIGNLIVE_SIGNUP_LINK** will be hidden on the Login page.  
Possible values:  
- **true** (default)  
- **false**

**REMOVE_FOR_ASSISTANCE**  
Boolean  
Flag that determines if the assistance phone number will be hidden in the top bar for third-party signers.  
Possible values:  
- **true** (default)  
- **false**

**ESL_TERMS_CONDITIONS_URL**  
String  
Terms and Conditions URL that appears in the application’s footer.  
**Note:** This parameter requires: (1) ESL_FOOTER_ON_LOGIN=true; (2) disabling of the *Disable Footer* account feature (see the Application Administrator’s Guide).  
Default:  
https://www.esignlive.com/landings/terms-and-conditions-service/
<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESL_PRIVACY_NOTICE_URL</td>
<td>string</td>
<td>Privacy Notice URL that appears in the application’s footer</td>
<td>Note: This parameter requires: (1) ESL_FOOTER_ON_LOGIN=true; (2) disabling of the Disable Footer account feature (see the Application Administrator’s Guide).</td>
</tr>
<tr>
<td>ESL_ONLINE_HELP_URL</td>
<td>string</td>
<td>OneSpan Sign’s Online Help URL that appears in the application’s footer</td>
<td>Note: This parameter requires: (1) ESL_SANDBOX=true; (2) ESL_FOOTER_ON_LOGIN=true; (3) disabling of the Disable Footer account feature (see the Application Administrator’s Guide).</td>
</tr>
<tr>
<td>ESL_SYSTEM_STATUS_URL</td>
<td>string</td>
<td>System Status URL that appears in the application’s footer</td>
<td>Note: This parameter requires: (1) ESL_FOOTER_ON_LOGIN=true; (2) disabling of the Disable Footer account feature (see the Application Administrator’s Guide).</td>
</tr>
<tr>
<td>PARAMETER</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
<td>VALUES</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>ESL_COMMUNITY_URL</td>
<td>string</td>
<td>OneSpan Sign’s Developer Community URL that appears in the application’s footer</td>
<td>Default: <a href="https://developer.esignlive.com/">https://developer.esignlive.com/</a></td>
</tr>
<tr>
<td>ESL_HOMEPAGE_URL</td>
<td>string</td>
<td>OneSpan Sign’s homepage URL that appears in the application’s footer</td>
<td>Default: <a href="https://www.onespan.com/sign">https://www.onespan.com/sign</a></td>
</tr>
</tbody>
</table>
| ESL_ENABLE_SENTRY         | boolean  | Flag that determines if Sentry will be enabled                              | Possible values:  
  • true  
  • false (default) |
| SENTRY_PUBLIC_DSN         | string   | Sentry’s public DSN (Data Source Name)                                      | No default |
| SENTRY_ENV                | string   | Sentry’s environment                                                       | Default: development |

**Note:** This parameter requires: (1) ESL_FOOTER_ON_LOGIN=true; (2) disabling of the Disable Footer account feature (see the Application Administrator’s Guide).

**Note:** This setting can help filter Sentry data.
<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOOTER_ON_LOGIN</td>
<td>boolean</td>
<td>Flag that determines if a footer with complete application information will appear on the Login page</td>
<td>Possible values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• false (default)</td>
</tr>
<tr>
<td>ESL_COOKIE_POLICY_NAME</td>
<td>string</td>
<td>Unique cookie name for your Cookie Policy</td>
<td>Default: esl-cookie-consent</td>
</tr>
<tr>
<td>ESL_COOKIE_POLICY_EXPIRES</td>
<td>string</td>
<td>Number of milliseconds from now when the Cookie Policy will expire</td>
<td>Default: 31536000000</td>
</tr>
<tr>
<td>PRIVATE_SETTINGS: Application Related</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APP_NAME</td>
<td>string</td>
<td>Name of the application</td>
<td>Default: OneSpan Sign</td>
</tr>
<tr>
<td>BUILD_DATE</td>
<td>string</td>
<td>Timestamp that will be output by sysinfo</td>
<td>A default is provided with each on-premises release.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: This information is provided when sysinfo is accessed.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Possible values:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• development</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• production (default)</td>
</tr>
<tr>
<td>RACK_ENV</td>
<td>string</td>
<td>Environment in which the Classic User Experience will run. This setting enables the code to be optimized for its environment.</td>
<td>Note: The value development enables debugging libraries. The value production should always be used when deploying in a Production environment.</td>
</tr>
</tbody>
</table>
## PRIVATE SETTINGS: Backend

<table>
<thead>
<tr>
<th><strong>PARAMETER</strong></th>
<th><strong>TYPE</strong></th>
<th><strong>DESCRIPTION</strong></th>
<th><strong>VALUES</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>API_HOSTNAME</td>
<td>string</td>
<td>Hostname needed to reach the API of the OneSpan Sign Application Backend</td>
<td>The possible values are any valid hostname. Default: localhost</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> For more information, see Step 6.</td>
</tr>
<tr>
<td>API_PORT</td>
<td>integer</td>
<td>Port needed to reach the API of the OneSpan Sign Application Backend</td>
<td>The possible values are any valid port number. Default: 443</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> For more information, see Step 6.</td>
</tr>
<tr>
<td>API_PROTOCOL</td>
<td>string</td>
<td>Web protocol used by the OneSpan Sign Application Backend server</td>
<td>Default: https</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> If you want to use SSL, specify the value https. <strong>Note:</strong> For more information, see Step 6.</td>
</tr>
<tr>
<td>BACKEND_API_VERSION</td>
<td>string</td>
<td>Version of the API of the OneSpan Sign Application Backend to which the Classic User Experience will connect</td>
<td>Default: 11.0.3</td>
</tr>
<tr>
<td>REST_PROXY_TIMEOUT</td>
<td>string</td>
<td>Number of seconds that a call to the OneSpan Sign Application Backend API will hang before timing out</td>
<td>Default: 240</td>
</tr>
</tbody>
</table>

## PRIVATE SETTINGS: Application Monitoring

<table>
<thead>
<tr>
<th><strong>PARAMETER</strong></th>
<th><strong>TYPE</strong></th>
<th><strong>DESCRIPTION</strong></th>
<th><strong>VALUES</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>NEW_RELIC_LICENSE_KEY</td>
<td>string</td>
<td>Your license key for New Relic</td>
<td>No default</td>
</tr>
<tr>
<td>NEW_RELIC_APP_NAME</td>
<td>string</td>
<td>Name that New Relic will use to identify your application</td>
<td>No default</td>
</tr>
</tbody>
</table>
Uninstalling the Classic User Experience

To uninstall the *Classic User Experience*:

1. Uninstall Ruby by running the following commands:
   
   ```
   gem uninstall bundler
   gem uninstall passenger
   gem uninstall rack
   gem uninstall rack-protection
   gem uninstall rack-test
   gem uninstall rake
   gem uninstall rdoc
   gem uninstall test-unit
   ```

2. Remove Apache configuration files by running the following commands:
   
   ```
   rm -rf /etc/httpd/conf.d/eSLFrontEnd.vhost.conf
   rm -rf /etc/httpd/conf.d/eSLMSC.vhost.conf
   rm -rf /etc/httpd/conf.d/eSLEvidence.vhost.conf
   ```

3. Restart HTTPD services.

4. Remove installed directories by running the following commands:
   
   ```
   rm -rf /opt/eSLFrontEnd/
   rm -rf /opt/eSLMSC/
   rm -rf <installation folder>/release/
   ```

   This procedure ends here.

Deploying the New User Experience

The *New User Experience* is the Application UX for senders and administrators that became available in *OneSpan Sign 6.1*. For sender and administrator actions, this experience represents an enhancement over the *Classic User Experience*. Its use is described in the *New Application User’s Guide*.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEW_RELIC_AGENT_ENABLED</td>
<td>boolean</td>
<td>Flag that determines if the New Relic agent will be used</td>
<td>• true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• false (default)</td>
</tr>
</tbody>
</table>

© 2020 - OneSpan North America Inc.  DG-OSS-72 - All rights reserved.  206
This section describes:

- Installing the New User Experience (page 207)
- Uninstalling the New User Experience (page 217)

## Installing the New User Experience

### Prerequisites

- Your environment meets all relevant requirements in Chapter 2.

  We **strongly** recommend that you install the *OneSpan Sign Application Frontend* on its own dedicated machine with a standard RHEL/CentOS 6 or 7 installation.

- The installer of the *OneSpan Sign Application Frontend* does not currently support automatically configuring 1-way or 2-way SSL connections.

- The procedure Installing Multiple Application Components (page 181) has been performed.

- If you had a prior installation of the New User Experience, or if you have an existing one, ensure that you have removed any associated services (e.g., pm2-init, esl-front-end).

- The procedure Installing the Classic User Experience (page 196) has been performed. To make the New User Experience work, the Classic User Experience must not only be installed, it must also be configured appropriately. We strongly recommend that you install the New User Experience on the same server as the Classic User Experience. If you do, one possible way of appropriately configuring the Classic User Experience is to insert the following lines in the Apache configuration file called `/etc/httpd/conf.d/eSLFrontEnd.vhost.conf`:

  ```
  RewriteRule ^/$ https://<frontend:port>/login
  RewriteRule ^/login(.*) https://<frontend:port>/a/login$1
  RewriteRule ^/error(.*) https://<frontend:port>/a/error$1
  ProxyPass         /a http://<new-ui>:3000/a
  ProxyPassReverse  /a http://<new-ui>:3000/a
  RewriteRule ^/a/(.*)$ - [L]
  RewriteRule ^/a/api/(.*)$ http(s)://<backend:port>/aws/rest/services/$1
  ```

  In the above lines:

  - `<frontend:port>` is a placeholder for the server name and port number associated with the *OneSpan Sign Application Frontend*.
  - `<backend:port>` is a placeholder for the server name and port number associated with the *OneSpan Sign Application Backend*.
  - `<new-ui>` is a placeholder for the server name associated with the *New User Experience*.
  - `/a` is a placeholder for the "eSignLive Basename".
**Action**

**To install and configure the New User Experience:**

1. Locate the file "esignlive-sender-ui-release_<version #>.zip" provided by OneSpan. Here <version #> is a placeholder for a version number (e.g., 1.9.1).

2. On the machine where you want to install the New User Experience, run the following commands:

   ```
   unzip esignlive-sender-ui-release_<version #>.zip
   cd release
   ./install.sh
   ```

3. Do one of the following:
   - If you are using RHEL/CentOS 7, copy the file "app_env.env" from the directory 'release' to the directory '/opt/esignlive/sender-ui/'.
   - If you are using RHEL/CentOS 6, copy the file "app_env.sh" from the directory 'release' to the directory '/opt/esignlive/sender-ui/'.

4. Do one of the following:
   - If you are using RHEL/CentOS 7, in the directory "/opt/esignlive/sender-ui/" edit the file "app_env.env" by appropriately specifying the parameters in Table 15-14. For example:

     ```
     NODE_ENV=production
     ESL_BASENAME=/uiux
     ESL_API_HOSTNAME=localhost
     ESL_API_PORT=8443
     ESL_API_PROTOCOL=https
     ESL_API_ALLOW_INSECURE=true
     # NODE_EXTRA_CA_CERTS=
     ```

     If you are using HTTPS (ESL_API_PROTOCOL=https), you must uncomment the last line above (by removing the #), and use it to specify the location of a PEM file that will be used as a trust store to communicate with the Application Server where the OneSpan Sign Application Backend is installed.

   - If you are using RHEL/CentOS 6, in the directory "/opt/esignlive/sender-ui/", edit the file "app_env.sh" by appropriately specifying the EXPORT parameters in Table 15-14. For example:

     ```
     export NODE_ENV=production
     export ESL_BASENAME=/uiux
     export ESL_API_HOSTNAME=localhost
     export ESL_API_PORT=8443
     export ESL_API_PROTOCOL=https
     ```
export ESL_API_ALLOW_INSECURE=true
# export NODE_EXTRA_CA_CERTS=

If you are using HTTPS (export ESL_API_PROTOCOL=https), you must uncomment the last line above (by removing the #), and use it to specify the location of a PEM file that will be used as a trust store to communicate with the Application Server where the OneSpan Sign Application Backend is installed.

Table 15-14’s PUBLIC SETTINGS do not contain sensitive information, so they are exposed to the browser. The PRIVATE SETTINGS do contain sensitive information, and so are never exposed to the browser.

5. A new service called sender-ui should now exist. Start this service by doing one of the following:
   • If you are using RHEL/CentOS 7, run the following command:
     systemctl start sender-ui
   • If you are using RHEL/CentOS 6, run the following command:
     service sender-ui start

This procedure ends here.

Table 15-14: Environment variables for the New User Experience

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESL_BASENAME</td>
<td>path</td>
<td>Path appended to the hostname</td>
<td>The possible values are any valid path that is prefixed with a forward slash &quot;/&quot;.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For example, <strong>uiux</strong> in:</td>
<td><strong><a href="http://apps.e-signlive.com/uiux">http://apps.e-signlive.com/uiux</a></strong> Default: /uiux</td>
</tr>
</tbody>
</table>
| ESL_SANDBOX             | boolean | Flag that determines if the environment is OneSpan Sign Sandbox | Possible values:  
|                         |       |                                                      | • **true**  
|                         |       |                                                      | • **false** (default) |
| ESL_UI_SWITCH_URL       | string | Path or fully qualified URL needed to reach the Classic User Experience | Default: /packages/inbox |
| ESL_ENABLE_LIVECHAT     | boolean | Flag that determines if Salesforce’s LiveAgent functionality is on | Possible values:  
|                         |       |                                                      | • **true**  
<p>|                         |       |                                                      | • <strong>false</strong> (default) |</p>
<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESL_LIVEAGENT_AGENT_ID</td>
<td>string</td>
<td>ID of a Salesforce LiveAgent agent</td>
<td>Note: This parameter requires ESL_ENABLE_LIVECHAT = true. Default: 57350000000KzWx</td>
</tr>
<tr>
<td>ESL_LIVEAGENT_DEPLOYMENT_ID</td>
<td>string</td>
<td>ID of a Salesforce LiveAgent deployment</td>
<td>Note: This parameter requires ESL_ENABLE_LIVECHAT = true. Default: 57250000000KzJH</td>
</tr>
<tr>
<td>ESL_LIVEAGENT_ORG_ID</td>
<td>string</td>
<td>ID of a Salesforce LiveAgent organization</td>
<td>Note: This parameter requires ESL_ENABLE_LIVECHAT = true. Default: 00D300000000IEw</td>
</tr>
<tr>
<td>ESL_LIVEAGENT_URL</td>
<td>string</td>
<td>URL of Salesforce’s LiveAgent application</td>
<td>Note: This parameter requires ESL_ENABLE_LIVECHAT = true. Default: <a href="https://d.la1-cl-dfw.salesforceliveagent.com/chat">https://d.la1-cl-dfw.salesforceliveagent.com/chat</a></td>
</tr>
<tr>
<td>ESL_ENABLE_RECAPTCHA</td>
<td>boolean</td>
<td>Flag that determines if the reCAPTCHA plugin will be used</td>
<td>Possible values: • true • false (default)</td>
</tr>
<tr>
<td>RECAPTCHA_SITE_KEY</td>
<td>string</td>
<td>Public site key of the reCAPTCHA application</td>
<td>Note: This parameter requires: (1) ESL_ENABLE_RECAPTCHA = true; (2) a value for RECAPTCHA_SECRET_KEY. No default For help with this parameter, consult this FAQ.</td>
</tr>
<tr>
<td>PARAMETER</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
<td>VALUES</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>ESL_ENABLE_FEEDBACK_FORM</td>
<td>boolean</td>
<td>Flag that determines if the feedback form for the New User Experience will be accessible</td>
<td>Possible values: • true • false (default)</td>
</tr>
<tr>
<td>ESL_ENABLE_GOOGLE_ANALYTICS</td>
<td>boolean</td>
<td>Flag that determines if Google Analytics will be enabled</td>
<td>Possible values: • true • false (default)</td>
</tr>
<tr>
<td>GOOGLE_ANALYTICS_UA</td>
<td>string</td>
<td>User account for Google Analytics</td>
<td>Default: UA-2950310-12</td>
</tr>
<tr>
<td>ESL_RELEASE_HIGHLIGHTS_URL</td>
<td>string</td>
<td>Content of the What's new link on the Login page of the New User Experience</td>
<td>No default</td>
</tr>
<tr>
<td>ESL_RELEASE_VERSION</td>
<td>string</td>
<td>Marketing release version that appears in the user account's drop-down menu</td>
<td>No default</td>
</tr>
<tr>
<td>ESL_SIGNUP_URL</td>
<td>string</td>
<td>URL of the Sign up link accessible from the Login page of the New User Experience</td>
<td>No default</td>
</tr>
</tbody>
</table>

**Note:** This parameter requires: (1) ESL_FOOTER_ON_LOGIN=true; (2) disabling of the Disable Footer account feature (see the Application Administrator’s Guide).
<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESL_TERMS_CONDITIONS_URL</td>
<td>string</td>
<td>Terms and Conditions URL that appears in the application’s footer</td>
<td>Default: <a href="https://www.esignlive.com/landings/terms-and-conditions-service/">https://www.esignlive.com/landings/terms-and-conditions-service/</a></td>
</tr>
<tr>
<td>ESL_PRIVACY_NOTICE_URL</td>
<td>string</td>
<td>Privacy Notice URL that appears in the application’s footer</td>
<td>Default: <a href="https://www.esignlive.com/landings/privacy-notice/">https://www.esignlive.com/landings/privacy-notice/</a></td>
</tr>
<tr>
<td>ESL_ONLINE_HELP_URL</td>
<td>string</td>
<td>OneSpan Sign’s Online Help URL that appears in the application’s footer</td>
<td>Default: <a href="https://docs.eesignlive.com/">https://docs.eesignlive.com/</a></td>
</tr>
</tbody>
</table>

**Note:** This parameter requires: (1) ESL_FOOTER_ON_LOGIN=true; (2) disabling of the Disable Footer account feature (see the Application Administrator’s Guide).
### Parameter Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESL_SYSTEM_STATUS_URL</td>
<td>string</td>
<td>System Status URL that appears in the application’s footer</td>
<td>Default: <a href="https://trust.esignlive.com/">https://trust.esignlive.com/</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This parameter requires: (1) ESL_FOOTER_ON_LOGIN=true; (2) disabling of the Disable Footer account feature (see the Application Administrator’s Guide).</td>
<td></td>
</tr>
<tr>
<td>ESL_COMMUNITY_URL</td>
<td>string</td>
<td>OneSpan Sign’s Developer Community URL that appears in the application’s footer</td>
<td>Default: <a href="https://developer.esignlive.com/">https://developer.esignlive.com/</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This parameter requires: (1) ESL_FOOTER_ON_LOGIN=true; (2) disabling of the Disable Footer account feature (see the Application Administrator’s Guide).</td>
<td></td>
</tr>
<tr>
<td>ESL_HOMEPAGE_URL</td>
<td>string</td>
<td>OneSpan Sign’s homepage URL that appears in the application’s footer</td>
<td>Default: <a href="https://www.onespan.com/sign">https://www.onespan.com/sign</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This parameter requires: (1) ESL_FOOTER_ON_LOGIN=true; (2) disabling of the Disable Footer account feature (see the Application Administrator’s Guide).</td>
<td></td>
</tr>
<tr>
<td>PARAMETER</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
<td>VALUES</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ESL_ENABLE_SENTRY</td>
<td>boolean</td>
<td>Flag that determines if Sentry will be enabled</td>
<td>Possible values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• false (default)</td>
</tr>
<tr>
<td>SENTRY_PUBLIC_DSN</td>
<td>string</td>
<td>Sentry’s public DSN (Data Source Name)</td>
<td>No default</td>
</tr>
<tr>
<td>SENTRY_ENV</td>
<td>string</td>
<td>Sentry’s environment</td>
<td>Default: development</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: This setting can help filter Sentry data.</td>
<td></td>
</tr>
<tr>
<td>ESL_FOOTER_ON_LOGIN</td>
<td>boolean</td>
<td>Flag that determines if a footer with complete application information will</td>
<td>Possible values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>appear on the Login page</td>
<td>• true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• false (default)</td>
</tr>
<tr>
<td>ESL_COOKIE_POLICY_NAME</td>
<td>string</td>
<td>Unique cookie name for your Cookie Policy</td>
<td>Default: esl-cookie-consent</td>
</tr>
<tr>
<td>ESL_COOKIE_POLICY_EXPIRES</td>
<td>string</td>
<td>Number of milliseconds from now when the Cookie Policy will expire</td>
<td>Default: 31536000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>PRIVATE SETTINGS: Application Related</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APP_VERSION</td>
<td>string</td>
<td>Version of the application that will be output by <code>sysinfo</code></td>
<td>A default is provided with each on-premises release.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: This information is provided when <code>sysinfo</code> is accessed.</td>
<td></td>
</tr>
<tr>
<td>APP_TIMESTAMP</td>
<td>string</td>
<td>Timestamp that will be output by <code>sysinfo</code></td>
<td>A default is provided with each on-premises release.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: This information is provided when <code>sysinfo</code> is accessed.</td>
<td></td>
</tr>
</tbody>
</table>
### Deploying the New User Experience

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
</table>
| NODE_ENV        | string   | The environment in which the New User Experience will run. This setting enables the code to be optimized for its environment.                                                                                | Possible values:  
• development  
• production (default) |

#### PRIVATE SETTINGS: Protocols & Certificates

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
</table>
| ESL_WEB_PROTOCOL| string   | Web protocol used by the OneSpan Sign Application Frontend server                                                                                                                                            | Possible values:  
• http (default) — For this value, the application will use ESL_NODE_PORT.  
• https — For this value: (1) the application will use ESL_SECURE_PORT; (2) you must specify SSL_KEY_PATH & SSL_CERT_PATH. |
| ESL_NODE_PORT   | integer  | Port used by the OneSpan Sign Application Frontend server if ESL_WEB_PROTOCOL=http                                                                                                                                 | The possible values are any valid port number.  
Default: 3000 |
| ESL_SECURE_PORT | integer  | Port used by the OneSpan Sign Application Frontend server if ESL_WEB_PROTOCOL=https                                                                                                                                          | Default: 3443 |
| SSL_KEY_PATH    | string   | Full path of the Private Key file used by SSL                                                                                                                                                               | Valid values are any file path that leads to a Private Key file.  
Note: Use this parameter only if you want to enable SSL with the New User Experience. No default |
### SSL_CERT_PATH

**Type:** string

*Full path of the SSL certificate file.*

**Description:** Use this parameter only if you want to enable SSL with the New User Experience.

**Values:** Valid values are any file path that leads to an SSL certificate file.

**Default:** No default

### PRIVATE SETTINGS: Backend

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESL_API_HOSTNAME</td>
<td>string</td>
<td>Hostname needed to reach the API of the OneSpan Sign Application Backend</td>
<td>The possible values are any valid hostname.</td>
</tr>
<tr>
<td>ESL_API_PORT</td>
<td>integer</td>
<td>Port needed to reach the API of the OneSpan Sign Application Backend</td>
<td>The possible values are any valid port number.</td>
</tr>
<tr>
<td>ESL_API_PROTOCOL</td>
<td>string</td>
<td>Web protocol used by the OneSpan Sign Application Backend server</td>
<td>If you do want to use SSL, specify the value https.</td>
</tr>
</tbody>
</table>
| ESL_API_ALLOW_INSECURE| boolean      | A flag that determines if an insecure proxy will be used to access the OneSpan Sign Application Backend | Possible values:
|                       |              | • true (default)                                                            | • false                                      |
|                       |              | **Warning:** Production environments should always use a managed certificate issued by a third-party provider such as VeriSign. |

### PRIVATE SETTING: Log Format

| MORGAN_LOG_FORMAT     | string       | Format used by the Morgan Logger                                           | The default is:                              |
|                       |              |                                                                             | :date[iso] INFO                              |
|                       |              |                                                                             | [:method] :url                                |
|                       |              |                                                                             | :remote-addr - :remote-user                  |
To uninstall the New User Experience:

1. Remove the Application directory.
2. Uninstall Node LTS.

The node.js component is typically removed by a superuser running the following command: `npm remove -g pm2`

This procedure ends here.

Deploying the Trusted Service Provider

The Trusted Service Provider is an Application module that enables signers to provide their own signing certificate, and thus to avoid installing a signing certificate within OneSpan Sign’s environment.

This section describes:

- Installing the Trusted Service Provider (page 218)
• Uninstalling the Trusted Service Provider (page 228)

Installing the Trusted Service Provider

Prerequisites

• Your environment meets all relevant requirements in Chapter 2.
• External Signer Verification (e.g., Validated ID, Asseco, DIGIPASS) must be enabled for your account. The Application Administrator’s Guide describes how to do this.

Action

To install the Trusted Service Provider:

1. If you are using a Load Balancer on the Trusted Service Provider’s server or servers:
   a. Install nginx on the Load Balancer’s host machine.
   b. Create a new file called upstreams-remote-signing.conf, and give it something like the following content:

```
Upstream remote_signing_stream {
    server ${remote-signing-host}:3000;
}
```

The above "server" command’s first argument is the URL of the Trusted Service Provider’s host server.

That command’s second argument is the Trusted Service Provider’s host port.

   c. Add this new file to the directory
      /etc/nginx/upstreams/includes.

   d. Create a new file called main-rewriterules-remote-signing.conf, and give it the following content:

```
location /remote-sign/ {
    include /etc/nginx/conf.d/proxy.conf;
    proxy_pass https://remote_signing_stream/;
}
```
e. Add this new file to the directory `/etc/nginx/sites-available/main-rewriterules`.

f. Open the file `/etc/nginx/sites-available/locations/locations.conf`.

g. Add the following line to that file:

```
"include /etc/nginx/sites-available/main-rewriterules/main-rewriterules-remote-signing.conf;"
```

h. Save and close that file.

2. Unzip the `install.tar.gz` file provided by OneSpan.

   If you have an RPM Package Manager, the tar file will be inside it.

A folder called `RS_Install` will be created in the location where you unzip the `tar` file.

3. In the `RS_Install` folder, locate the general configuration file called `settings.json`. In that file, configure for your application environment all the parameters in Table 15-15 except: (1) must-use-secret-manager; (2) enc-ssl-key-passphrase; (3) enc-jwe-secret.

4. In the `RS_Install` folder, locate the configuration file for Asseco signing called `asseco-settings.json`. In that file, configure for your application environment all the parameters in Table 15-16 except: (1) enc-asseco-client-id; (2) enc-asseco-client-secret.

5. In the `RS_Install` folder, locate the configuration file for Validated ID signing called `validatedId-settings.json`. In that file, configure for your application environment all the parameters in Table 15-17 except: (1) enc-validatedId-client-id; (2) enc-validatedId-apiKey; (3) enc-validatedId-scope.

6. Store the following six "secret parameters" needed for this application's environment in a safe place (e.g., in a storage vault, in an encrypted location, on paper; but not on disk):

   a. SSL password for the SSL certificate associated with `ssl-cert` in Table 15-15
   b. Asseco Client ID for accessing Asseco
   c. Asseco Client Secret for use in Asseco
   d. Validate Client ID for accessing Validated ID
   e. Validate Api Key for use in Validated ID
   f. JWE secret for decoding JWE

   The Installer will use the above six parameters in this order when the next step runs the script `Install.sh`. 
7. In the RS_Install folder, locate the file Install.sh. Then install the Remote Signing service by running the following command, using the six ordered arguments from the previous step:

```
./Install.sh arg1 arg2 arg3 arg4 arg5 arg6
```

If you are running this command as the root user, the Installer will put the generated file remote_signing.service in the directory /usr/lib/systemd/system.

8. If you ran the previous step’s command as a non-root user, you must:
   a. Log on as the root user, and manually copy the generated file remote_signing.service into the directory /usr/lib/systemd/system.
   b. Run the following command as the root user:

```
systemctl daemon-reload
```

9. Verify that the RS_Install folder now contains the following three generated files:

   • enc-settings.json
   • enc-asseco-settings.json
   • enc-validatedId-settings.json

As these filenames suggest, these output files correspond to the three input JSON files you configured in Step 3, Step 4 and Step 5. These output files contain encrypted values of the six secret parameters in Step 6.

10. Install and run the redis service. To do so for a personal deployment:
    a. Install the package epel-release.
    b. Install the package redis, using the appropriate version of that package for your RHEL/Centos machine.
    c. Run the following command:

```
systemctl start redis
```

11. Start the Remote Signing service by running the following command:

```
systemctl start remote_signing
```

12. Once this service is running, the log file configured via the first four parameters in Table 15-15 will start filling with information about your service and its connection to redis. If an error code appears in that file, you may want to consult the code descriptions in Table 15-18.

This procedure ends here.
### Table 15-15: Parameters in settings.json

<table>
<thead>
<tr>
<th>KEY</th>
<th>DESCRIPTION / VALUE</th>
<th>TYPE</th>
<th>ENCRYPTED</th>
<th>OPTIONAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>log-level</td>
<td>Possible values:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• ERROR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• WARNING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• INFO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DEBUG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DEBUG1-4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: DEBUG4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>log-maxsize</td>
<td>Maximum size of the log file (in kB)</td>
<td>int</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default: 50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>log-maxcount</td>
<td>Maximum number of stored log files before they begin to be overwritten</td>
<td>int</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default: 10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>log-path</td>
<td>A valid absolute path to the directory where the log file will be stored. The path should not include the filename.</td>
<td>string</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Example: /tmp/</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: ./logs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>callback-url</td>
<td>URL of the Remote Signing service</td>
<td>string</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Example: <a href="https://esl-fake.silanis.com/remote-signing">https://esl-fake.silanis.com/remote-signing</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>esl-frontend-url</td>
<td>URL of OneSpan Sign's frontend</td>
<td>string</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Example: <a href="https://preview.esignlive.com">https://preview.esignlive.com</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KEY</td>
<td>DESCRIPTION / VALUE</td>
<td>TYPE</td>
<td>ENCRYPTED</td>
<td>OPTIONAL</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td>check-esl-cookie</td>
<td>Flag that determines if the validity of cookies from the OneSpan Sign Application Frontend will be checked</td>
<td>bool</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Possible values:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• false</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• true</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>One of the above values must be specified (there is no default).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>redis-host</td>
<td>The redis server's hostname or IP address</td>
<td>string</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Default: RedisHost</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>redis-port</td>
<td>The redis server's port number</td>
<td>int</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Default: 6379</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>redis-timeout</td>
<td>The redis server's connection timeout in seconds</td>
<td>int</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Default: 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>redis-key-ttl</td>
<td>How long a redis key will live (in seconds)</td>
<td>int</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default: 3600</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ssl-cert</td>
<td>Absolute path of the certificate needed to enable SSL</td>
<td>string</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default: &lt;empty string&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ssl-key</td>
<td>Absolute path of the key needed to enable SSL</td>
<td>string</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default: &lt;empty string&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>enc-ssl-key-passphrase</td>
<td>Pass phrase for the key in the previous row</td>
<td>string</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default: &lt;empty string&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>port</td>
<td>The port number that will be used by Remote Signing</td>
<td>int</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default: 3000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 15-16: Parameters in asseco-settings.json

<table>
<thead>
<tr>
<th>KEY</th>
<th>DESCRIPTION / VALUE</th>
<th>TYPE</th>
<th>ENCRYPTED</th>
<th>OPTIONAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>proxy-url</td>
<td>Proxy URL and port</td>
<td>string</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Example: <a href="http://proxy.esign-live.com:1080">http://proxy.esign-live.com:1080</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: &lt;empty string&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>must-use-secret-manager</td>
<td>Flag that determines if One-Span Sign’s Secret Manager will be used</td>
<td>bool</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Possible values:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 1: Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 0 (default): No — This value must be specified if you have an on-premises deployment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cookie-ttl</td>
<td>How long the Remote Signing cookie will live (in seconds)</td>
<td>int</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default: 3600</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>common ssl-name</td>
<td>DNS name used by the SSL certificate</td>
<td>string</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Default: &lt;empty string&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>enc-jwe-secret</td>
<td>Pass phrase for decoding JWE</td>
<td>string</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Default: &lt;empty string&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>api-url</td>
<td>URL of OneSpan Sign’s REST API</td>
<td>string</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Default: &lt;empty string&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max-requests-per-child</td>
<td>Maximum number of requests allowed per Remote Signing child process</td>
<td>int</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default: 1000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>enc-asseco-client-id</td>
<td>Client ID provided by Asseco</td>
<td>string</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default: assecoclientid</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 15-17: Parameters in validatedID-settings.json

<table>
<thead>
<tr>
<th>KEY</th>
<th>DESCRIPTION / VALUE</th>
<th>TYPE</th>
<th>ENCRYPTED</th>
<th>OPTIONAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>enc-asseco-client-secret</td>
<td>Client secret provided by Asseco</td>
<td>string</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default: assecoclientsecret</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>asseco-url</td>
<td>Entry point for Asseco’s signing service</td>
<td>string</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default: <a href="https://model.simplysign.webnotarius.pl">https://model.simplysign.webnotarius.pl</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>asseco-oauth</td>
<td>Asseco Oauth service API</td>
<td>string</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default:/idp/oauth2.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>asseco-task</td>
<td>Asseco’s signing service task API</td>
<td>string</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default: /card/v1/certificates/tasks</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>asseco-revoke</td>
<td>Asseco’s access token revocation API</td>
<td>string</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default: /cas/oauth2.0/revoke</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>asseco-sign</td>
<td>Asseco’s signing service API</td>
<td>string</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default:/card/v1/cards/%s/certificates/signature</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>asseco-verification-type</td>
<td>Asseco’s verification type (e.g., POC = proof of concept)</td>
<td>string</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default: AssecoPOC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>validatedId-url</td>
<td>Entry point of Validated ID’s signing service</td>
<td>string</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Default: <a href="https://pre-vidsignercloudmgmt.validate-did.com">https://pre-vidsignercloudmgmt.validate-did.com</a></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 15-18: Error Codes

<table>
<thead>
<tr>
<th>CODE</th>
<th>TEXT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>RS_NO_ERROR</td>
<td>There are no errors.</td>
</tr>
<tr>
<td>1</td>
<td>RS_RUNTIME_ERROR</td>
<td>There is a runtime error.</td>
</tr>
<tr>
<td>2</td>
<td>RS_NULL_PTR_ERROR</td>
<td>The null pointer has been reached.</td>
</tr>
<tr>
<td>3</td>
<td>RS_UNEXPECTED_ERROR</td>
<td>An unknown error occurred.</td>
</tr>
<tr>
<td>4</td>
<td>RS_UNSUPPORTED_CA</td>
<td>The provided ID for the Certificate Authority is not supported.</td>
</tr>
<tr>
<td>5</td>
<td>RS_FAIL_TO_OPEN_CONFIG_FILE</td>
<td>Cannot load the file settings.json.</td>
</tr>
<tr>
<td>6</td>
<td>RS_NO_COOKIE</td>
<td>The Trusted Service Provider did not receive a cookie from the Front End.</td>
</tr>
<tr>
<td>7</td>
<td>RS_URL_NO_FOUND</td>
<td>The Trusted Service Provider's URL was not found.</td>
</tr>
<tr>
<td>CODE</td>
<td>TEXT</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>11</td>
<td>STRU_STRING_NOT_FOUND</td>
<td>Target string is not found in <code>StringUtil</code>.</td>
</tr>
<tr>
<td>12</td>
<td>STRU_IO_ERROR</td>
<td>Encountered an Input/Output error in <code>StringUtil</code>.</td>
</tr>
<tr>
<td>13</td>
<td>STRU_INVALID_INPUT_VALUE</td>
<td>Invalid input value</td>
</tr>
<tr>
<td>14</td>
<td>RS_WRONG_CA_SETTING</td>
<td>The <code>tid</code> parameter's value does not match any supported remote Certificate Authority. This is probably a configuration issue.</td>
</tr>
<tr>
<td>21</td>
<td>ESL_SESSION_ID_UNAVAILABLE</td>
<td>Cannot obtain a session ID for OneSpan Sign.</td>
</tr>
<tr>
<td>22</td>
<td>ESL_TRANSACTION_GUID_UNAVAILABLE</td>
<td>Cannot obtain a transaction GUID for OneSpan Sign.</td>
</tr>
<tr>
<td>23</td>
<td>ESL_TRANSACTION_IS_COMPLETE</td>
<td>The transaction has been completed, so it can no longer be modified.</td>
</tr>
<tr>
<td>24</td>
<td>ESL_CURRENT_TRANS_HAS_NO_APPROVAL</td>
<td>Cannot find any approval in the current transaction.</td>
</tr>
<tr>
<td>25</td>
<td>ESL_CURRENT_DOC_NOT_FOUND_IN_TRANS</td>
<td>Cannot find any document in the current transaction.</td>
</tr>
<tr>
<td>26</td>
<td>ESL_DOCUMENT_GUID_UNAVAILABLE</td>
<td>Cannot find the given document's GUID.</td>
</tr>
<tr>
<td>27</td>
<td>ESL_API_URL_UNAVAILABLE</td>
<td>Cannot attain the URL for OneSpan Sign's API.</td>
</tr>
<tr>
<td>28</td>
<td>ESL_NO_VERIFICATION_REQ_TOKEN</td>
<td>The expected <code>Verification Request Token</code> was not found in the OneSpan Sign request.</td>
</tr>
<tr>
<td>31</td>
<td>JSON_KEY_NOT_FOUND</td>
<td>The desired key was not found in the provided JSON.</td>
</tr>
<tr>
<td>32</td>
<td>JSON_NULL_PTR</td>
<td>The desired key points to a null object.</td>
</tr>
<tr>
<td>33</td>
<td>JSON_NULL_VALUE</td>
<td>The desired value is null for the provided key.</td>
</tr>
<tr>
<td>34</td>
<td>JSON_NO_ARRAY</td>
<td>The provided JSON is not an array.</td>
</tr>
<tr>
<td>35</td>
<td>JSON_WRONG_VALUE_TYPE</td>
<td>The provided JSON had a value of the wrong type.</td>
</tr>
<tr>
<td>36</td>
<td>JSONJwtFailedToCreateJWKKey</td>
<td>Failed to create a key from the provided 'secret'.</td>
</tr>
<tr>
<td>37</td>
<td>JSONJwtFailedToImportPayload</td>
<td>Failed to import the provided JWT token. The input token is probably corrupted.</td>
</tr>
<tr>
<td>38</td>
<td>JSONJwtFailedToDecrypt</td>
<td>JWE decryption failed.</td>
</tr>
<tr>
<td>39</td>
<td>JSONJwtDecryptedTokenIsEmpty</td>
<td>JWE decryption did not return an error, but the result is an empty string.</td>
</tr>
<tr>
<td>CODE</td>
<td>TEXT</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>------</td>
<td>------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>40</td>
<td>JSON_JWT_DECRYPTED_TOKEN_IS_INVALID</td>
<td>JWE decryption succeeded, but the result is malformed.</td>
</tr>
<tr>
<td>41</td>
<td>XML_PARSE_ERROR</td>
<td>Failed to parse the provided XML.</td>
</tr>
<tr>
<td>42</td>
<td>XML_NULL_PTR</td>
<td>The provided XML was a null pointer.</td>
</tr>
<tr>
<td>43</td>
<td>XML_NULL_VALUE</td>
<td>The provided XML was a null value.</td>
</tr>
<tr>
<td>51</td>
<td>ASSECO_FAIL_TO_GET_API_KEY</td>
<td>Cannot get the API key from Asseco.</td>
</tr>
<tr>
<td>52</td>
<td>ASSECO_FAIL_TO_GET_SIGNER_CERT</td>
<td>Cannot get the signer certificate from Asseco.</td>
</tr>
<tr>
<td>53</td>
<td>ASSECO_FAIL_TO_SIGN_HASH</td>
<td>Failed to sign a hash from Asseco.</td>
</tr>
<tr>
<td>54</td>
<td>ASSECO_FAIL_TO_GET_RESOURCE</td>
<td>Failed to get a resource from Asseco.</td>
</tr>
<tr>
<td>55</td>
<td>ASSECO_FAIL_TO_GET_PIN</td>
<td>The Asseco client failed to receive the PIN.</td>
</tr>
<tr>
<td>56</td>
<td>ASSECO_WRONG_PIN</td>
<td>The Asseco client has the wrong PIN.</td>
</tr>
<tr>
<td>57</td>
<td>ASSECO_SERVICE_PENDING</td>
<td>The Asseco service is not responding yet, but is pending.</td>
</tr>
<tr>
<td>61</td>
<td>VALID_FAIL_TO_GET_AUTH_TOKEN</td>
<td>Cannot get an authentication token from Validated Id.</td>
</tr>
<tr>
<td>62</td>
<td>VALID_FAIL_TO_GET_SIGNER_CERT</td>
<td>Cannot get a signer certificate from Validated Id.</td>
</tr>
<tr>
<td>63</td>
<td>VALID_FAIL_TO_GET_SIGNER_CERT_NO</td>
<td>Wrong certificate number</td>
</tr>
<tr>
<td>64</td>
<td>VALID_FAIL_TO_SIGN_HASH</td>
<td>Failed to sign a hash from Validated Id.</td>
</tr>
<tr>
<td>71</td>
<td>REDIS_FAIL_TO_OPEN_SESSION</td>
<td>The Redis server failed to open the session.</td>
</tr>
<tr>
<td>72</td>
<td>REDIS_UNEXPECTED_INPUT_TYPE</td>
<td>The Redis server received an unexpected input value.</td>
</tr>
<tr>
<td>73</td>
<td>REDIS_SET_OP_ERROR</td>
<td>The Redis server encountered an operations failure.</td>
</tr>
<tr>
<td>74</td>
<td>REDIS_REPLY_RETURN_ERROR</td>
<td>The Redis server returned an error.</td>
</tr>
<tr>
<td>75</td>
<td>REDIS_REPLY_RETURN_NO_RESULT</td>
<td>The Redis server returned no result.</td>
</tr>
<tr>
<td>80</td>
<td>CRYPTO_SUCCESS</td>
<td>The ECC was successfully encrypted.</td>
</tr>
<tr>
<td>81</td>
<td>CRYPTO_ERR_SYM_CIPHER</td>
<td>The ECC’s cryptographic encoding key was incompatible with its decoding key.</td>
</tr>
<tr>
<td>82</td>
<td>CRYPTO_ERR_BAD_PARAMETER</td>
<td>The ECC’s secret key contained a bad parameter.</td>
</tr>
<tr>
<td>83</td>
<td>CRYPTO_EMPTY_MSG</td>
<td>The ECC contained an empty message.</td>
</tr>
</tbody>
</table>
Uninstalling the Trusted Service Provider

To uninstall the Trusted Service Provider:

1. Stop the remote_signing.service.
2. Remove the RS_Install folder.

This procedure ends here.

Deploying the OneSpan Sign Document Converter

The OneSpan Sign Document Converter is an Application module that converts a document from DOC or ODT format to PDF format.

This section describes:

- Installing the OneSpan Sign Document Converter (page 228)
- Uninstalling the OneSpan Sign Document Converter (page 229)

Installing the OneSpan Sign Document Converter

Prerequisites

- The procedure Installing Multiple Application Components (page 181) has been performed.
- Your environment meets all relevant requirements in Chapter 2.

We recommend that you install the OneSpan Sign Document Converter on its own dedicated machine.

<table>
<thead>
<tr>
<th>CODE</th>
<th>TEXT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>84</td>
<td>CRYPTO_RSA_PUBLIC_ENCRYPT_FAIL</td>
<td>The ECC’s public key was not encrypted.</td>
</tr>
<tr>
<td>85</td>
<td>CRYPTO_PUB_KEY_ALGORITHM_NO_FOUND</td>
<td>The ECC’s public key’s encryption algorithm was not found.</td>
</tr>
<tr>
<td>86</td>
<td>CRYPTO_WRONG_PUB_KEY_ALGORITHM</td>
<td>The ECC’s public key was using the wrong encryption algorithm.</td>
</tr>
<tr>
<td>87</td>
<td>CRYPTO_EXTRACT_PUB_KEY_FAIL</td>
<td>The ECC’s public key could not be extracted.</td>
</tr>
</tbody>
</table>
Action

To install and configure the OneSpan Sign Document Converter:

1. Create a Document Converter directory on the machine from which you will run the converter – for example:
   - c:\esignlive\document-converter

2. Copy into that directory all Document Converter files provided by OneSpan.

3. Right-click the Command Prompt, and select More > Run as Administrator.

4. Inside the file esl-document-converter.xml, configure the Java memory setting (the argument \-Xmx) in accordance with the amount of RAM available on your instance. A good rule-of-thumb is to allocate 50% of the available RAM to Java (e.g., if the instance has 4 GB of available RAM, allocate 2048 MB to Java).

5. Navigate to the directory c:\esignlive\document-converter, and run the following command:
   - esl-document-converter.exe install
   This will install the Windows Service that will run the converter.

6. To start the service, run the following command:
   - esl-document-converter.exe start

7. Further configure the converter by specifying appropriate values for the key "esl.converter.aspose" in Table 18-1.

8. Verify the installation by opening the Windows Services view. You should see the service eSignLive Document Service in the state Running.

This service can be managed from Windows Services or from the command line. If you want to stop the service from the command line, in Administrator mode run the following command: esl-document-converter.exe stop

This procedure ends here.

Uninstalling the OneSpan Sign Document Converter

To uninstall the OneSpan Sign Document Converter:

1. Stop the service by running the following command in Administrator mode:
   - esl-document-converter.exe stop

2. Uninstall the service by running the following command in Administrator mode:
   - esl-document-converter.exe uninstall

3. Remove the installation directory that was created during the procedure Installing the OneSpan Sign Document Converter (page 228).
Installing Connectors for the Application

On-premises deployments of OneSpan Sign support all "connectors" in the Optional Application Components section of Table 1-1 except eSignLive for Box.

Before a particular connector can receive a callback from OneSpan Sign, a component called the Connector Framework must be installed.

This section describes:

- Installing the Connector Framework (page 230)
- Installing Particular Connectors (page 233)

Installing the Connector Framework

The Connector Framework is a standalone component that enables a connection between individual connectors and OneSpan Sign. It runs on:

- Any server where the OneSpan Sign Application Backend (esl.ear) runs
- Any server where OneSpan Sign BackOffice (backoffice.ear) runs

This section describes:

- Installing the Framework on CentOS7 (page 230)
- Installing the Framework on CentOS6 (page 232)

Table 2-2 lists several Operating Systems besides CentOS 7 and CentOS 6. This guide does not yet describe how to install the Connector Framework on them. However: (1) that documentation is being prepared; (2) the installation steps are very similar for all Operating Systems.

Installing the Framework on CentOS7

Prerequisites

- The procedure Installing Multiple Application Components (page 181) has been performed.
- Your environment meets all relevant requirements in Chapter 2.

Action

To install the Connector Framework on CentOS 7, do the following on each server where the files esl.ear and backoffice.ear are deployed:

1. Create the following directory:
   
   /opt/esignlive/connectors-application

2. Into that directory, copy the following file provided by OneSpan:
   
   esignlive-connectors-framework-<version>.tar.gz
3. Extract that file by running the following command:
   
   ```bash
   tar -zxvf esignlive-connectors-framework-<version>.tar.gz
   ```
   
   Ensure that the files connectors-application-<version>.jar, log.xml and logback.xml are present in the directory /opt/esignlive/connectors-application.

4. Extract the file `application.yml` into the following directory:
   
   /opt/esignlive/connectors-application

5. Create a Symbolic Link by running the following command:
   
   ```bash
   ln -sf /opt/esignlive/connectors-application/connectors-application-<version>.jar /opt/esignlive/connectors-application/connectors-application.jar
   ```

6. Ensure that the following logging configuration is present in the file /opt/esignlive/connectors-application/application.yml:

   ```yaml
   logging:
       config: /opt/esignlive/connectors-application/log.xml
   ```

7. Go to the directory /etc/systemd/system.

8. Open the service file for the Connector Framework by running the following command:

   ```bash
   vi connectors-application.service
   ```

9. Paste the following lines into the service file:

   ```bash
   [Unit]
   Description=Connectors Application
   [Service]
   ExecStart=/opt/esignlive/connectors-application/connectors-application.jar
   SuccessExitStatus=143
   [Install]
   WantedBy=multi-user.target
   ```

10. Save and close the service file.

11. Run the following command to change the service file's permissions:

   ```bash
   sudo chmod 0755 connectors-application.service
   ```

   If you already used root to log in to CentOS 7, you can skip the next step.

12. Run the following command to change the service file’s owner to root:
sudo chown root:root connectors-application.service

13. Load the service for the Connector Framework by running the following command:

   systemctl daemon-reload

14. Run the following command to start the service for the Connector Framework:

   systemctl start connectors-application.service

15. Check the status of the service by running the following command:

   systemctl status connectors-application.service

16. Verify that the service started successfully by ensuring that there are no errors in the following log file:

   /var/log/connectors-application/connectors-application-<date>.log

   This procedure ends here.

### Installing the Framework on CentOS6

**Prerequisites**

- The procedure Installing Multiple Application Components (page 181) has been performed.
- Your environment meets all relevant requirements in Chapter 2.

**Action**

To install the Connector Framework on CentOS 6, do the following on each server where the files esl.ear and backoffice.ear are deployed:

1. Create the following directory:

   /opt/esignlive/connectors-application

2. Into that directory, copy the following file provided by OneSpan:

   esignlive-connectors-framework-<version>.tar.gz

3. Extract that file by running the following command:

   tar -zxvf esignlive-connectors-framework-<version>.tar.gz

   Ensure that the files connectors-application-<version>.jar, log.xml and logback.xml are present in the directory /opt/esignlive/connectors-application.

4. Extract the file application.yml into the following directory:

   /opt/esignlive/connectors-application

5. Create a Symbolic Link by running the following commands:

   cd /opt/esignlive/connectors-application
ln -s -T connectors-application-<version>.jar connectors-application.jar

6. Ensure that the following logging configuration is present in the file
   /opt/esignlive/connectors-application/application.yml:

   ```yaml
   logging:
     config: /opt/esignlive/connectors-application/log.xml
   ```

7. Create another Symbolic Link by running the following commands:

   ```bash
   cd /etc/init.d
   ln -s -T /opt/esignlive/connectors-application/connectors-application.jar connectors-application
   ```

8. Start and then stop the Connector Framework service by running the
   following commands:

   ```bash
   cd /opt/esignlive/connectors-application
   service connectors-application start
   service connectors-application stop
   ```

9. Verify that the service started successfully by ensuring that there are
   no errors in the following log file:

   ```bash
   /var/log/connectors-application/connectors-application-<date>.log
   ```

   This procedure ends here.

---

**Installing Particular Connectors**

**Prerequisites**

- The procedure Installing the Connector Framework (page 230) has
  been performed.
- Your environment meets all relevant requirements in Chapter 2.

**Action**

To deploy a particular connector:

- Follow the instructions in the connector’s deployment guide.

Links to online and PDF versions of that guide can be found here:

[https://docs.esignlive.com/content/a_esignlive/product_guides.htm](https://docs.esignlive.com/content/a_esignlive/product_guides.htm)

This procedure ends here.

---

**Installing the Personal Certificate Client**

The *Personal Certificate Client* is an Application module used to sign documents
with a personal digital certificate.
The Personal Certificate Client secures communication with the system via a mechanism based on digital signatures, hashes, timestamps, and two-way authentication.

To work with your particular configuration, the Personal Certificate Client’s setup program must be customized to reflect the certificates used in your environment. OneSpan will provide this customization, but you must first contact Technical Support (support@onespan.com; 1-855-MYESIGN).
# PART VI: OPTIONAL HARDWARE

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>16: Optional Hardware</td>
<td>236</td>
</tr>
</tbody>
</table>
CHAPTER 16: Optional Hardware

OneSpan Sign can work with a number of optional hardware components, but needs none of them in order to work properly.

To make use of those hardware components, see the following sections:

- Configuring a Load Balancer (page 236)
- Configuring a Proxy Server (page 236)
- Integrating a Hardware Security Module (page 237)

Configuring a Load Balancer

A horizontally scaled environment with a Load Balancer is strongly recommended.

The Load Balancer is an application that: (i) distributes workloads across multiple resources; (ii) serves as the system’s gatekeeper for all incoming requests from the Internet.

A Load Balancer should direct all Application-related traffic to the OneSpan Sign Application Frontend.

Essentially: (1) there should be only one URL exposed for the Application UI and API; (2) all incoming traffic on that base URL should be sent to the Apache server running the UI component. This Apache server will then determine if the traffic is for the UI or for the API, and forward accordingly.

Neither OneSpan Sign BackOffice nor the Admin Console should be exposed publicly.

Configuring a Proxy Server

The Proxy Server is the system’s gatekeeper for all outgoing traffic to the Internet. If you want to deploy a Proxy Server, configure it using the following procedure.

To configure a Proxy Server to work with the Application:

1. Navigate to the URL of your BackOffice installation. A Login screen appears.
2. Log in to BackOffice using your username and password. A Welcome screen appears.
3. Click Manage Resources. The Resources for all Accounts screen appears.
4. Edit the parameters in Table 16-1. The proxy.properties Resource Key is listed in the leftmost column. To edit the parameters for that key, click the Edit icon in its rightmost column. A new screen displays the key’s
parameters. When you’ve finished editing those parameters, click **Save**.

This procedure ends here.

**Table 16-1: Configuration parameters required for a Proxy Server**

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
<th>PLACEHOLDER VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Key = proxy.properties</td>
<td>Name of the host where the Proxy Server runs</td>
<td>Default: &lt;empty&gt;</td>
</tr>
<tr>
<td>host</td>
<td><strong>Note</strong>: All parameters in this table are for an HTTP Proxy Server.</td>
<td>Example: localhost2</td>
</tr>
<tr>
<td>port</td>
<td>Number of the port where the Proxy Server runs</td>
<td>Default: &lt;empty&gt;</td>
</tr>
<tr>
<td>nonProxyHosts</td>
<td>List of hosts of services that can be called directly by the Application without going through a Proxy Server. This list should include the PDF Document Engine and the OneSpan Sign Document Converter.</td>
<td>Default: &lt;empty&gt;</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: This parameter is relevant only if you have configured host and port.</td>
<td>Example (a pipe-separated list): server.01</td>
</tr>
</tbody>
</table>

**Integrating a Hardware Security Module**

*OneSpan Sign* can be used with a *Hardware Security Module* (HSM) that provides several benefits:

- It safeguards and manages digital keys for strong authentication.
- It provides cryptographic encoding/decoding of signer certificates.
- It isolates private keys from network attacks.

If you use an HSM, it will be located either on the Cloud or your organization’s premises. Thus just one of the following sections is relevant to your deployment:

- [Using an HSM Located on the Cloud](page 238)
- [Using an HSM Located On Premises](page 245)

⚠️ We currently support connecting only one HSM to each PDF Document Engine server environment, whether you’re installing on the Cloud or on-premises.
The HSM policy on FIPS must be set to *FIPS-enabled* mode.

**Using an HSM Located on the Cloud**

If you use an HSM that is located on the Cloud, it must be provided by either Gemalto or Amazon. Thus just one of the following sections is relevant to your deployment:

- Gemalto’s Cloud HSM-7 (DPOD) (page 238)
- Amazon Web Services (AWS) (page 241)

**Gemalto’s Cloud HSM-7 (DPOD)**

**Prerequisites**

- Your environment meets all relevant requirements in Chapter 2.
- Your PDF Document Engine is running a Linux Operating System.

**Action**

**To integrate a PDF Document Engine with a Gemalto HSM-7 Service Client:**

1. If the PDF Document Engine is running, stop it.
2. Log in to the following site:
   
   https://esignlive.na.market.dpondemand.io

3. Add a new service (select **Digital Signing**).
4. Select the option to create a Service Client. This will create a *zip* file.
5. Download the resultant file `Centos7_client.zip`, and unzip it on your PDF Document Engine server that will host the HSM-7 Service Client.
6. In the directory `<Service Client installation folder>` where you unzipped the file in the previous step, run the following command:

   ```bash
tar xvf cvclient-min.tar
   ```

7. In the `<Service Client installation folder>` directory, run the following commands:

   ```bash
   source ./setenv
   ./bin/64/lunacm
   ```

A default slot ID should appear. We recommend that you use this ID in the next step.
8. Run the following command:

```bash
lunacm> slot set -slot <desired slot ID>
```

9. Run the following command:

```bash
lunacm> par init -label <desired partition name>
```

10. In response to successive prompts that appear:
   a. Type `proceed`.
   b. Enter a desired password for the Partition Security Officer.
   c. Enter a desired domain name (e.g., `esignlive`).

11. Successively run the following five commands:
   - ```bash
      lunacm> role login -name po <the password defined in the previous step>
   ```
     
     In the previous command, `po` is an abbreviation for Partition Security Officer.
   - ```bash
      lunacm> role init -name co <desired password>
   ```
     
     In the previous command, `co` is an abbreviation for Crypto Officer.
   - ```bash
      lunacm> role logout
   ```
   - ```bash
      lunacm> role login -name co <the Crypto Officer password defined earlier in this step>
   ```
   - ```bash
      lunacm> role changepw -name co <change the CO password>
   ```

12. Run the following command:

```bash
lunacm> exit
```

13. From the directory `<Service Client installation folder>`:
   a. Open the file `Chrystoki.conf`.
   b. Add the following line to that file:

   ```
   LibPath64=<path to libCrystoki2.so>
   ```

   The path to `libCrystoki2.so` appears inside the file `Chrystoki.conf` in the line above where `libUNIX64` is defined.

14. Make a copy of the file `Chrystoki.conf` (leaving the original where it is), and put that copy in the directory `/etc`. 
15. In the directory `<Service Client installation folder> /bin/64`, find the tool `cmu`.

16. Run the following three commands:
   - `s./cmu generatekeypair -modulusBits=2048 -publicExp=65537 -sign=T -verify=T -labelPublic="public.key" -labelPrivate="private.key"
   - `./cmu list
   - `./cmu requestCert -sha256withrsa -publichandle=<public key handle id> -privatehandle=<private key handle id> -C=CA -S=QC -L=Montreal -O="Silanis" -OU="RD" -CN="<certname>" -outputFile=<certname>.csr

17. Give your Certificate Authority certificate provider the file `<certname>.csr` from the previous step. In return, that provider will give you your purchased Signer Certificate and its certificate chain (i.e., its Trusted Certificates & Intermediate Certificates).

18. Ensure that all Intermediate Certificates are in one file, and that all Trusted Certificates are in another file. Place those files and the Signer Certificate in the Remote Properties Repository for the PDF Document Engine properties associated with each tenant (see Table 11-1 and Table 11-2).

19. In view of the previous step, update the parameters `TrustedCertificates`, `IntermediateCertificates`, and `SignerCertificate` in Table 11-2.

20. Obtain from Gemalto the `sautil` tool for your Linux Operating System.

21. From the directory where you installed `sautil`, run the following command:
   - `./sautil -o -a 0:RSA -f privatekeyhandle.pem -s <slotid> -v -q -p <changed co password>

22. Put the file `privatekeyhandle.pem` in the Remote Properties Repository for the PDF Document Engine properties associated with each tenant. Then make the parameter `CPSFile` in Table 11-2 point to this file, ensuring that you follow the note in that table regarding the formatting of this parameter.

   If you are in a non-tenant environment, specify these parameters using the file "DocumentEngineSettings.props". If you are in a tenant environment, do so using the Admin Console.

   If you are in a non-tenant environment, specify `CPSFile` using the file "DocumentEngineSettings.props". If you are in a tenant environment, do so using the Admin Console.
23. Assign the parameter CPSPassword in Table 11-2 the base64-encoded version of the "partition CO" password created in Step 11. It will become the HSM Signing Partition’s password.

If you are in a non-tenant environment, specify CPSPassword using the file "DocumentEngineSettings.props". If you are in a tenant environment, do so using the Admin Console. For details, see Step 4 and Step 5 of the procedure Configuring a PDF Document Engine.

24. Start the PDF Document Engine with Apache. It should now be connected to the Gemalto HSM-7 Service Client.

This procedure ends here.

Amazon Web Services (AWS)

Prerequisites

- Your environment meets all relevant requirements in Chapter 2.
- Your PDF Document Engine is running a Linux Operating System.

Action

To integrate a PDF Document Engine with an AWS CloudHSM Service Client:

1. If the PDF Document Engine is running, stop it.
2. Open the AWS CloudHSM console page at:

   https://console.aws.amazon.com/cloudhsm/

   If you don't already have an AWS Cloud account, create an IAM user AWS Cloud account by clicking the Create Account button.

3. Create a cluster that can contain one or more HSMs by following the instructions at:

   https://docs.aws.amazon.com/cloudhsm/latest/userguide/create-cluster.html

4. Create an HSM by following the instructions at:

   https://docs.aws.amazon.com/cloudhsm/latest/userguide/create-hsm.html

5. Initialize the cluster created in Step 3 by doing the following:
   a. Open the AWS CloudHSM console page at:

      https://console.aws.amazon.com/cloudhsm/

   b. Next to the cluster, click Initialize.

6. Use the links that will be created to download the following four documents: (1) Cluster CSR; (2) HSM Certificate; (3) AWS Certificate; (4) Manufacturer Certificate.
7. Obtain the root certificate for the AWS cloudhsm from:
   https://docs.aws.amazon.com/cloudhsm/latest/userguide/root-certificate.html and choose AWS_CloudHSM_Root-G1.zip

8. Obtain the HSM Manufacturer Certificate from:
   https://www.cavium.com/LS/TAmanuCert/
   Then select Download Certificate.

9. Assign your CSR file the name <cluster ID>A_ClusterCSR.csr. Then ensure that it and the other files you just downloaded are in the same directory on the machine that runs the PDF Document Engine.

10. Sign the CSR by running the following openssl commands from the console of the machine that runs the PDF Document Engine:
   • openssl genrsa -aes256 -out customerCA.key 2048
     The preceding command creates a private key.
   • openssl req -new -x509 -days 3652 -key customerCA.key -out customerCA.crt
     The preceding command creates a self-signed certificate.

11. Optional: If you want to create a cluster with multiple HSMs:
   a. Run the following openssl command, which creates the signer certificate for your HSM cluster:
      openssl x509 -req -days 3652 -in <cluster ID>_ClusterCsr.csr -CA customerCA.crt -CAkey customerCA.key -CAcreateserial -out <cluster ID>_CustomerHsmCertificate.crt
   b. Initialize the HSM cluster by doing the following:
      A. Open the AWS CloudHSM console page at:
         https://console.aws.amazon.com/cloudhsm/
      B. Next to the HSM cluster, click Initialize.
      c. On the Download certificate signing request page, click Next.
      d. On the Sign certificate signing request (CSR) page, choose Next.
      e. On the Upload the certificates page:
         A. Next to Cluster certificate, click Upload file.
         B. Select the file <clusterID>_CustomerHsmCertificate.crt that was created in Step 11a.
      f. On the Upload the certificates page:
         A. Next to Issuing certificate, click Upload file.
         B. Select the file customerCA.crt that was created in Step 10.
12. By running the following commands, begin installing the CloudHSM Client on the machine that runs the PDF Document Engine:

```bash
wget https://s3.amazonaws.com/cloudhsmv2-software/cloudhsm-clientlatest.x86_64.rpm
sudo yum install -y ./cloudhsm-client-latest.x86_64.rpm
```

13. Copy the file `customerCA.crt` from Step 10 into the directory `/opt/cloudhsm/etc`.

```
The directory /opt/cloudhsm/ is the installation directory for the CloudHSM Client created in Step 12.
```

14. Run the following command:

```bash
sudo /opt/cloudhsm/bin/configure -a <IP address>
```

Here `<IP address>` is the IP address of the HSM created in Step 4. That address appears in the information about your HSM cluster on the AWS CloudHSM console page, https://console.aws.amazon.com/cloudhsm/.

15. Activate your HSM cluster by following the instructions at:

https://docs.aws.amazon.com/cloudhsm/latest/userguide/activate-cluster.html

16. To enable the HSM cluster to see the machine where your PDF Document Engine is installed, launch an Amazon EC2 Client instance by following the instructions at:


17. You began installing the CloudHSM Client in Step 12. To complete its installation, run the following commands on the machine where your PDF Document Engine is installed:

```bash
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-pkcs11-latest.el7.x86_64.rpm
sudo yum install -y ./cloudhsm-client-pkcs11-latest.el7.x86_64.rpm
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-dyn-latest.el7.x86_64.rpm
sudo yum install -y ./cloudhsm-client-dyn-latest.el7.x86_64.rpm
export n3fips_password=<CO username>:<CO password>
```

In the preceding command, the username and password placeholders refer to the CO User Account created in Step 15.
18. To create the private-key handle and the Certificate Signer Request (CSR) that will be used to create the Signer Certificate for the PDF Document Engine, run the following commands on the machine where your PDF Document Engine is installed:

   `./openssl genrsa -engine cloudhsm -out aws_hsm_key.pem 2048`

   `./openssl req -engine cloudhsm -new -key aws_hsm_key.pem -out aws_hsm.csr`

19. The file "aws_hsm.csr" was created in the last command of the preceding step. Give a copy of that file to your Certificate Authority certificate provider. In return, they will give you your purchased Signer Certificate and its certificate chain (i.e., its Trusted Certificate & Intermediate Certificates).

20. Ensure that all Intermediate Certificates are in one file, and that all Trusted Certificates are in another file. Place those files and the Signer Certificate in the Remote Properties Repository for the PDF Document Engine properties associated with each tenant (see Table 11-1 and Table 11-2).


   If you are in a non-tenant environment, specify these parameters using the file "DocumentEngineSettings.props". If you are in a tenant environment, do so using the Admin Console.

22. Specify the parameter CPSFile that appears in Table 11-2. The relevant file is "aws_hsm_key.pem", which was created in Step 18.

   If you are in a non-tenant environment, specify CPSFile using the file "DocumentEngineSettings.props". If you are in a tenant environment, do so using the Admin Console.

23. Specify the parameter CPSPassword that appears in Table 11-2. You must specify its value using the format <CO username>:<CO password>, where <CO username> and <CO password> are defined in Step 17.

   If you are in a non-tenant environment, specify CPSPassword using the file "DocumentEngineSettings.props". If you are in a tenant environment, do so using the Admin Console. For details, see Step 4 and Step 5 of the procedure Configuring a PDF Document Engine (page 129).
24. On the machine where your PDF Document Engine is installed, start the *CloudHSM Client* by running the following command:

    sudo /opt/cloudhsm/bin/cloudhsm_client
    /opt/cloudhsm/etc/cloudhsm_client.cfg

We highly recommend that you run the preceding command as a service.


    This procedure ends here.

**Using an HSM Located On Premises**

If you use an HSM located on your organization’s premises, it must be one of the following products provided by Gemalto:

- Gemalto’s On-Premises HSM-7 (page 245)
- Gemalto’s On-Premises HSM-6 (page 246)

**Gemalto’s On-Premises HSM-7**

**Prerequisites**

- Your environment meets all relevant requirements in Chapter 2.
- Your PDF Document Engine is running a Linux Operating System.

**Action**

The procedures for using Gemalto’s on-premises HSM-7 are identical to those in the section Gemalto’s On-Premises HSM-6 (page 246), except that:

- Gemalto’s On-Premises HSM-6 (page 246) contains instructions for both Linux and Windows. However, we have not yet tested HSM-7 on Windows. Thus we do not yet support HSM-7 on Windows.

- The firmware versions for on-premises HSM-7 differ from those for on-premises HSM-6 (see Table 2-4).

- For HMS-6, the file Chrystoki.conf contains the following lines:

  ```
  EngineLunaCA3 = {
    LibPath64=/usr/lib/Cryptoki_64.so;
    EngineInit=1:1:1;
  }
  ```

  For HSM-7, these lines should be replaced by the following:

  ```
  Chrystoki2 = {
    LibUNIX64=/opt/gemsigning/64/libCryptoki2.so;
    LibPath64=/opt/gemsigning/64/libCryptoki2.so;
  }
  ```

- For HMS-6, the following sample commands appear in Step 4 of
Integrating on Linux (page 250):

```
./sautil -o -s 1 -i 1:1 -p <MySigningPartitionPassword>
./sautil -a 0:RSA -f MyPrivateKeyHandle.pem -s 1 -i 1:1
./sautil -c -s 1 -i 1:1
```

For HSM-7, these commands should be replaced by the following:

```
./sautil -o -a 0:RSA -f MyPrivateKeyHandle.pem -s 1 -v -p <MySigningPartitionPassword>
```

- Step 6 of Integrating on Linux (page 246) should be replaced with Step 7, Step 8, Step 9, Step 10, and Step 11 of Gemalto’s Cloud HSM-7 (DPOD) (page 238). The only qualification here is that you should ignore the first of the two commands in Step 7.

- Instead of performing both parts of Step 7 of Integrating on Linux (page 246), you should perform just its second part (Step 7b).

Gemalto’s On-Premises HSM-6

To integrate Gemalto’s on-premises HSM-6 with the system, sequentially perform the following procedures:

- Integrating the HSM with the Luna SA Client (page 246)
- Integrating the HSM, Luna Client, and Doc Engine (page 249)

These procedures are followed by a discussion of:

- A Known Limitation (page 259)
- A Known Issue with Luna K-6 Firmware (page 260)

**Integrating the HSM with the Luna SA Client**

Your PDF Document Engine must run either a Linux or Windows Operating System (see Table 2-4). Accordingly, perform the appropriate one of the following procedures:

- Integrating on Linux (page 246)
- Integrating on Windows (page 248)

**Integrating on Linux**

**Prerequisites**

- Your PDF Document Engine is running a Linux Operating System.
- A SafeNet Luna SA HSM has been installed on your network.

SafeNet Luna SA is now known as SafeNet Network HSM.

- The Luna SA Client has been installed and configured on each PDF Document Engine server. The supported versions of that software are specified in Table 2-4.
**Action**

**To integrate your HSM with the Luna SA Client on Linux:**

1. On the server where the Luna SA Client is installed (i.e., the PDF Document Engine server), go to the directory `<Luna Client installation folder>/bin`.

2. Generate a Luna SA Client certificate by doing the following:
   a. Run the following command on the Luna SA Client:
      
      ```
      ./vtl createCert -n <Luna Client IP address>
      ```
   b. Place the resultant PEM file (e.g., myClient1.pem) in the directory `<Luna Client installation folder>/cert/client`.

   Subsequent steps assume that the name of the certificate file generated here is "myClient1.pem".

3. Install the HSM's server certificate on the Luna SA Client by doing the following:
   a. Run the following command on the Luna SA Client:
      
      ```
      ./scp admin@<HSM IP address>:server.pem server.pem
      ```
      
      You will be prompted to type the password of the HSM Admin.
   b. Move the exported "server.pem" file to the directory `<Luna Client installation folder>/cert/server`.

4. Export the file "myClient1.pem" from Step 2b to the HSM by running the following command on the Luna SA Client:

      ```
      ./scp <Luna Client installation folder>/cert/client/myClient1.pem admin@<HSM IP address>:
      ```

      You will be prompted to type the password of the HSM Admin.

5. Register the HSM with the Luna SA Client by running the following command on the Luna SA Client:

      ```
      ./vtl addServer -n <HSM IP address> -c <Luna Client installation folder>/cert/server/server.pem
      ```

6. Create an HSM signing partition, and assign the Luna SA Client to that partition by doing the following:
   a. Log in to the HSM as its `admin`.
   b. Run the following 3 commands on the HSM:
      
      ```
      hsm:>client register -client <Luna Client name> -hostname MyClient1
      ```
      
      Here `<Luna Client name>` can be any name that refers to the Luna SA Client.

      ```
      hsm:>partition create -partition <partition name>
      ```
Using an HSM Located On Premises

7. Verify the success of the preceding steps by doing the following:
   a. Confirm that client information is successfully stored on the HSM partition by running the following command on the HSM:
      
      ```
hsm:>client show -client <Luna Client name>
      ```
   b. Confirm that a "slot" and a partition have been assigned to the Luna SA Client on the HSM by running the following command on the Luna SA Client:
      
      ```
./vtl verify
      ```

This procedure ends here.

**Integrating on Windows**

**Prerequisites**
- Your PDF Document Engine is running a Windows Operating System.
- A SafeNet Luna SA HSM has been installed on your network.

*SafeNet Luna SA is now known as SafeNet Network HSM.*

- The Luna SA Client has been installed and configured on each PDF Document Engine server. The supported versions of that software are specified in Table 2-4.

**Action**

**To integrate your HSM with the Luna SA Client on Windows:**

1. On the server where the Luna SA Client is installed (i.e., the PDF Document Engine server), go to the directory `<Luna Client installation folder>`.

2. Generate a Luna SA Client certificate by doing the following:
   a. Run the following command on the Luna SA Client:
      
      ```
      vtl.exe createCert -n <Luna Client IP address>
      ```
   b. Place the resultant PEM file (e.g., `myClient1.pem`) in the directory `<Luna Client installation folder>/cert/client`.

   Subsequent steps assume that the name of the certificate file generated here is "myClient1.pem".

3. Install the HSM’s server certificate on the Luna SA Client by doing the following:
   a. Run the following command on the Luna SA Client:
Using an HSM Located On Premises

cscp.exe admin@<HSM IP address>:server.pem server.pem

You will be prompted to type the password of the HSM Admin.

b. Move the exported "server.pem" file to the directory <Luna Client installation folder>/cert/server.

4. Export the file "myClient1.pem" from Step 2b to the HSM by running the following command on the Luna SA Client:

cscp.exe <Luna Client installation folder>/cert/client/myClient1.pem admin@<HSM IP address>:

You will be prompted to type the password of the HSM Admin.

5. Register the HSM with the Luna SA Client by running the following command on the Luna SA Client:

vtl.exe addServer -n <HSM IP address> -c <Luna Client installation folder>/cert/server/server.pem

6. Create an HSM signing partition, and assign the Luna SA Client to that partition by doing the following:

   a. Log in to the HSM as its admin.

   b. Run the following 3 commands on the HSM:

   hsm:>client register -client <Luna Client name> -hostname MyClient1

   Here <Luna Client name> can be any name that refers to the Luna SA Client.

   hsm:>partition create -partition <partition name>

   hsm:>client assignPartition -client <Luna Client name> -partition <partition name>

7. Verify the success of the preceding steps by doing the following:

   a. Confirm that client information is successfully stored on the HSM partition by running the following command on the HSM:

   hsm:>client show -client <Luna Client name>

   b. Confirm that a "slot" and a partition have been assigned to the Luna SA Client on the HSM by running the following command on the Luna SA Client:

   vtl.exe verify

This procedure ends here.

Integrating the HSM, Luna Client, and Doc Engine

Your PDF Document Engine must run either a Linux or Windows Operating System (see Table 2-4). Accordingly, perform the appropriate one of the following procedures:
Integrating on Linux

Prerequisites

- Your PDF Document Engine is running a Linux Operating System.
- The HSM has been integrated with the Luna SA Client. See Integrating the HSM with the Luna SA Client (page 246).

Action

To integrate the HSM, Luna SA Client, and a Linux PDF Document Engine:

1. Put the file "Chrystoki.conf" in the folder /etc:

   ![Configuration file](image)

   The configuration file Chrystoki.conf comes with your installation of the Luna SA Client.

2. Add the following lines to the file "Chrystoki.conf":

   ```
   EngineLunaCA3 = {
   LibPath64=<full path to Cryptoki64.dll or Cryptoki_64.so>;
   EngineInit=<token slot ID>::<application ID major>::<application ID minor>;
   }
   ```

   In the preceding line: (1) <token slot ID> is the slot ID of the token to be used for authentication (default = 1); (2) <application id major/minor> are two 32-bit integers that identify the application this procedure is trying to create (i.e., this PDF Document Engine instance integrated with the HSM and Luna SA Client).

   Example

   ```
   EngineLunaCA3 = {
   LibPath64=/usr/lib/Cryptoki_64.so;
   EngineInit=1:1:1;
   }
   ```

3. If you plan to use Method 2 in Step 5, create a CSR (Certificate Signing Request) file from the HSM by running a suitable series of commands on the Luna SA Client. The following is one possible series of commands:

   ```
   ./cmu generatekeypair -modulusBits=2048
   -publicExp=65537 -sign=T -verify=T
   -labelPublic="public.key" -labelPrivate="private.key"
   ```

   ![Command](image)

   The preceding command assumes a 2048-bit key.

   2048 is the minimum number of bits required to comply with FIPS 140-2.
• ./cmu list

The preceding command enables you to view which handle IDs are used to store the public and private keys of the pair generated by the first command.

• ./cmu requestCert -sha256withrsa -publichandle=its handle ID> -privatehandle=its handle ID> -C=<country> -S=<state> -L=<city> -O="<company>" -OU="<group>" -CN="<email>" -outputFile =sha256withrsa.certificate.csr

The preceding command assumes that all signatures will be created using the SHA-256RSA algorithm. This is the minimum required to comply with FIPS 140-2. The command’s bracketed <placeholders> are self-explanatory.

NOTE: If you are in a testing environment rather than a production environment, you may be able to use self-signed certificates. If so, you can:

a. Replace the preceding command (.cmu requestCert ...) by the following command:

   ./cmu selfSign -publichandle=its handle ID> -privatehandle=its handle ID> -C=<country> -S=<state> -L=<city> -O="<company>" -OU="<group>" -CN="<email>"

b. Download the signer certificate using the following command:

   ./cmu export -handle=its handle ID> -output=selfsigned.pem

   For example:

   ./cmu export -handle=1 -output=selfsigned.pem

4. To extract from the HSM the PEM file which contains the private-key handle that will be used for signing, run the following commands with the tool sautil on a Linux machine that has SSH access to the HSM:

   • ./sautil -o -s <token slot ID> -i <application ID major>:<application ID minor> -p <password>
   • ./sautil -a 0:RSA -f <filename> -s <token slot ID> -i <application ID major>:<application ID minor>
   • ./sautil -c -s <token slot ID> -i <application ID major>:<application ID minor>

In the preceding commands: (1) <token slot ID> and <application id major/minor> are as in Step 2; (2) <password> is the required password of the slot; (3) <filename> is the desired name of an output PEM file that contains the extracted private-key handle.
For example (using slot 1):

- ./sautil -o -s 1 -i 1:1 -p <MySigningPartitionPassword>
- ./sautil -a 0:RSA -f MyPrivateKeyHandle.pem -s 1 -i 1:1
- ./sautil -c -s 1 -i 1:1

5. Create the following three PEM files that contain specific base-64-encoded certificates:

- A file that contains a signer certificate (aka a "leaf certificate")
- A file that contains the signer's root certificate (aka a "trusted certificate")
- A file that contains relevant intermediate certificates (but no self-signed ones)

You may create the file for intermediate certificates by concatenating multiple PEM files, thus storing all intermediate certificates in a single PEM file. Concatenation is not permitted for the signer or root certificates.

You can extract the above certificates using any of the following alternative methods:

- **Method 1**: If you are not in a production environment, you can use the commands in Step 3 to extract self-signed certificates from the HSM.

  To see a list of the HSM’s stored objects and their handle IDs, execute the Luna SA Client "cmu list" command.

- **Method 2**: Use an OpenSSL command-line tool that extracts certificates from a PFX file that you purchased from a Certificate Provider with the CSR file created in Step 3.

- **Method 3**: Use certificates installed on your computer, and extract them from your browser using its Base-64 encoded X.509 option.

6. Create a directory that will store the relevant certificates and private-key handle, and copy into it the PEM files from Step 4 and Step 5. For example, that directory could be:

  <Doc Engine installation folder>/pem

   Apache users must be able to access that directory.

7. Ensure that these parameters from Table 11-2 are configured as follows:

   - CPSFile = <Full path of the private-key handle>
   - CPSPassword = <Password of the HSM Signing Partition>
• **PEMPath** = <The directory created in Step 6>

   **PEMPath** is required only in a non-tenant environment.

• **TrustedCertificates** = <Name of the PEM file with the root certificate>

• **IntermediateCertificates** = <Name of the PEM file with all intermediate certificates>

• **SignerCertificate** = <Name of the PEM file with the signer certificate>

• **TimeStampServer/URI** = <timestamp-server URI>

• **CoreTrace** = DEBUG1

8. Log into the HSM using the Luna SA Client’s salogin tool by running the following command:

   ```
   ./salogin -o -s <token slot ID> -i <application ID major>:<application ID minor> -p <password>
   ```

For example:

   ```
   ./salogin -o -s 1 -i 1:1 -p <MySigningPartitionPassword>
   ```

   The password is required. The preceding command will execute without it, but the Luna Client will not be able to access the HSM.

   If for any reason you want to close the connection between the HSM and the Luna Client, use the command:

   ```
   ./salogin -c -s <token slot ID> -i <application ID major>:<application ID minor>
   ```

9. Configure the parameter **CPSPassword** in Table 11-2 to be the HSM Signing Partition’s password. If you are in a non-tenant environment, do so using the file "DocumentEngineSettings.props". If you are in a tenant environment, do so using the Admin Console. For details, see Step 5 of the procedure Configuring a PDF Document Engine (page 131).

10. Start the PDF Document Engine with Apache.

11. Verify that the above steps were successful by signing a document.

12. If you are having trouble registering your HSM with your Luna Client, you may want to try the following suggestions:

   a. To test the HSM’s ability to self-sign, execute the following command from the Luna Client (available on Luna Client v5.3.5 and higher):
Using an HSM Located On Premises

.b. To list all client objects created on your HSM, execute the following command from the Luna Client:

    ./cmu list

c. To list active time slots seen by the Luna Client on your HSM, execute the following command from the Luna Client:

    ./vtl verify

d. To list all slots of your HSM, execute the following command from the Luna Client:

    ./vtl listSlots

e. To list partitions, firmware, and the feature status of your HSM, execute the following command from the HSM:

    hsm:>hsm show

This procedure ends here.

Integrating on Windows

Prerequisites

- Your PDF Document Engine is running a Windows Operating System.
- The HSM has been integrated with the Luna SA Client. See Integrating the HSM with the Luna SA Client (page 246).

Action

To integrate the HSM, Luna SA Client, and a Windows PDF Document Engine:

1. Make a copy of the crystoki.ini file in your main Luna Client installation folder, and call it Chrystoki.conf. Both files should reside in the same folder.

   The configuration file crystoki.ini comes with your installation of the Luna SA Client.

2. Set the following environment variable:

   ChrystokiConfigurationPath= <folder that contains Chrystoki.conf>

3. Add the following lines to the top of the file "Chrystoki.conf":

   {EngineLunaCA3}
   LibPath64=<full path to Cryptoki64.dll or Cryptoki_64.so>;
   LibNT64=<full path to Cryptoki64.dll or Cryptoki_64.so>;

   You should consult Gemalto about the use of the preceding command.
Using an HSM Located On Premises

LibNT=<full path to Cryptoki64.dll or Cryptoki_64.so>;
EngineInit=<token slot ID>:<application ID major>:<application ID minor>;

In the preceding line: (1) <token slot ID> is the slot ID of the token to be used for authentication (default = 1); (2) <application id major/minor> are two 32-bit integers that identify the application this procedure is trying to create (i.e., this PDF Document Engine instance integrated with the HSM and Luna SA Client); (3) the above semicolons are needed; (4) LibPath64, LibNT64, and LibNT are absolute paths, not relative ones; (5) do not use quotation marks; (6) note the curly brackets in the first line.

Example

{EngineLunaCA3}
LibPath64=C:\Program Files\lunasa\lib\Cryptoki64.dll;
LibNT64=C:\Program Files\lunasa\lib\Cryptoki64.dll;
LibNT=C:\Program Files\lunasa\lib\Cryptoki64.dll;
EngineInit=1:1:1;

4. In the file crystoki.ini, ensure that LibNT64 and LibNT are defined in the same way as they’re defined in Chrystoki.conf. Both variables for both files should be in the section [Chrystoki2].

5. If you plan to use Method 2 in Step 7, create a CSR (Certificate Signing Request) file from the HSM by running a suitable series of commands on the Luna SA Client. The following is one possible series of commands:

• cmu.exe generatekeypair -modulusBits=2048
  -publicExp=65537 -sign=T -verify=T
  -labelPublic="public.key" -labelPrivate="private.key"

The preceding command assumes a 2048-bit key. 2048 is the minimum number of bits required to comply with FIPS 140-2.

• cmu.exe list

The preceding command enables you to view which handle IDs are used to store the public and private keys of the pair generated by the first command.

• cmu.exe requestCert -sha256withrsa -publichandle=<its handle ID> -privatehandle=<its handle ID> -C=<country> -S=<state> -L=<city> -O="<company>" -OU="<group>" -CN="<email>" -outputFile =sha256withrsa.certificate.csr

The preceding command assumes that all signatures will be created using the SHA-256RSA algorithm. This is the minimum required to comply with FIPS 140-2. The command’s bracketed <placeholders> are self-explanatory.
NOTE: If you are in a testing environment rather than a production environment, you may be able to use self-signed certificates. If so, you can:

a. Replace the preceding command (`./cmu requestCert ...`) by the following command:

```bash
cmu.exe selfSign -publichandle=<its handle ID>
-privatehandle=<its handle ID> -C=<country> -S=<state>
-L=<city> -O="<company>" -OU="<group>" -CN="<email>"
```

b. Download the signer certificate using the following command:

```bash
cmu.exe export -handle=<its handle ID> -output= selfsigned.pem
```

For example:

```
cmu.exe export -handle=1 -output=selfsigned.pem
```

6. To extract from the HSM the PEM file which contains the private-key handle that will be used for signing, run the following commands with the tool `sautil` on a Windows machine that has SSH access to the HSM:

- `sautil.exe -o -s <token slot ID> -i <application ID major>:<application ID minor> -p <password>
- `sautil.exe -a 0:RSA -f <filename> -s <token slot ID> -i <application ID major>:<application ID minor>
- `sautil.exe -c -s <token slot ID> -i <application ID major>:<application ID minor>

In the preceding commands: (1) `<token slot ID>` and `<application id major/minor>` are as in Step 3; (2) `<password>` is the password of the slot; (3) `<filename>` is the desired name of an output PEM file that contains the extracted private-key handle.

For example:

- `sautil.exe -o -s 1 -i 1:1 -p <MySigningPartitionPassword>
- `sautil.exe -a 0:RSA -f MyPrivateKeyHandle.pem -s 1 -i 1:1
- `sautil.exe -c -s 1 -i 1:1`

7. Create the following three PEM files that contain specific base-64-encoded certificates:

- A file that contains a signer certificate (aka a "leaf certificate")
- A file that contains the signer’s root certificate (aka a "trusted certificate")
- A file that contains relevant intermediate certificates (but no self-
signed ones)

You may create the file for intermediate certificates by concatenating multiple PEM files, thus storing all intermediate certificates in a single PEM file. Concatenation is not permitted for the signer or root certificates.

You can extract the above certificates using any of the following alternative methods:

- **Method 1**: If you are not in a production environment, you can use the commands in Step 5 to extract self-signed certificates from the HSM.

  To see a list of the HSM’s stored objects and their handle IDs, execute the Luna SA Client "cmu list" command.

- **Method 2**: Use an OpenSSL command-line tool that extracts certificates from a PFX file that you purchased from a Certificate Provider with the CSR file created in Step 5.

- **Method 3**: Use certificates installed on your computer (e.g., in the Windows Certificate Store), and extract them from your browser using its Base-64 encoded X.509 option (see the sample screen below).
8. Create a directory that will store the relevant certificates and private-key handle, and copy into it the PEM files from Step 6 and Step 7. For example, that directory could be:

<Doc Engine installation folder>/pem

Apache/IIS users must be able to access that directory.

9. Ensure that these parameters from Table 11-2 are configured as follows:

- CPSFile = <Full path of the private-key handle>
- CPSPassword = <Password of the HSM Signing Partition>
- PEMPath = <The directory created in Step 8>

PEMPath is required only in a non-tenant environment.

- TrustedCertificates = <Name of the PEM file with the root certificate>
- IntermediateCertificates = <Name of the PEM file with all intermediate certificates>
- SignerCertificate = <Name of the PEM file with the signer certificate>
- TimeStampServer/URI = <timestamp-server URI>
- CoreTrace = DEBUG1

For example, you might have:

- CPSFile = C:Program Files/Luna SA/Keyhandle.pem
- CPSPassword = <Password of the HSM Signing Partition>
- PEMPath = C:/Program Files/Silanis/PDFDocumentEngine/pem/
- TrustedCertificates = adobe.pem
- IntermediateCertificates = intermediate.pem
- SignerCertificate = signer.pem

10. Log into the HSM using the Luna SA Client’s salogin tool by running the following command:

```
salogin.exe -o -s <token slot ID> -i <application ID major>:<application ID minor> -p <password>
```

For example:

```
salogin.exe -o -s 1 -i 1:1 -p <MySigningPartitionPassword>
```

The password is required. The preceding command will execute without it, but the Luna Client will not be able to access the HSM.
11. Configure the parameter CPSPassword in Table 11-2 to be the HSM Signing Partition’s password. If you are in a non-tenant environment, do so using the file "DocumentEngineSettings.props". If you are in a tenant environment, do so using the Admin Console. For details, see Step 5 of the procedure Configuring a PDF Document Engine (page 131).

12. Start the PDF Document Engine with Apache or IIS.

13. Verify that the above steps were successful by signing a document.

14. If you are having trouble registering your HSM with your Luna Client, you may want to try the following suggestions:

   a. To test the HSM’s ability to self-sign, execute the following command from the Luna Client (available on Luna Client v5.3.5 and higher):

      ```
      cmu.exe verifyhsm
      ```

      ![Warning] You should consult Gemalto about the use of the preceding command.

   b. To list all client objects created on your HSM, execute the following command from the Luna Client:

      ```
      cmu.exe list
      ```

   c. To list active time slots seen by the Luna Client on your HSM, execute the following command from the Luna Client:

      ```
      vtl.exe verify
      ```

   d. To list all slots of your HSM, execute the following command from the Luna Client:

      ```
      vtl.exe listSlots
      ```

   e. To list partitions, firmware, and the feature status of your HSM, execute the following command from the HSM:

      ```
      hsm:>hsm show
      ```

This procedure ends here.

**A Known Limitation**

**The Problem**

An important limitation of the Luna Client is that if its connection to the HSM gets broken, any PDF Document Engine process that encounters the associated error
will not recover. To reestablish communication with the HSM, the PDF Document Engine must be restarted.

For an Apache installation, if the Luna Client’s connection to the HSM gets broken, at least one Apache process becomes “poisoned” in the sense that it will return the same associated OpenSSL error for each request, even if the connection is later restored.

For a PDF Document Engine error, normally:

- The Platform re-tries the operation.
- If very few Apache processes are “poisoned”, the system may continue working properly without noticeable issues (other than occasional error messages in the logs).
- If more than “very few” processes are poisoned, the system may sporadically fail to sign documents until the PDF Document Engine is restarted.

**Workaround for a Linux Installation**

To minimize the impact of this limitation, and to avoid having to restart the PDF Document Engine, if you have a Linux installation, we recommend using the parameter CriticalErrors described in Table 11-2.

When a PDF Document Engine process encounters an error that has been identified as ”critical” by the CriticalErrors parameter, then:

- That process will will exit immediately.
- If necessary (depending on the current load), Apache will spawn a fresh process in its place.

If you want to identify an HSM communication failure as a critical error, please use the following setting:

- CriticalErrors=66;8009108D;

**A Known Issue with Luna K-6 Firmware**

This section applies to any version of Luna K-6 firmware less than 6.24.3. The issue was fixed in version 6.24.3.

If the HSM becomes disconnected from the Luna Client (because of a network disruption, for example), the salogin feature may wrongly indicate that the HSM is still connected.

If you encounter this situation, close and reopen the connection using salogin. To recall how to do so, see:

- Step 8 if you are using Linux
- Step 10 if you are using Windows
# PART VII: CONFIGURING THE SYSTEM

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>17: Configuring the Platform</td>
<td>262</td>
</tr>
<tr>
<td>18: Configuring the Application</td>
<td>265</td>
</tr>
</tbody>
</table>
CHAPTER 17: Configuring the Platform

This chapter assumes that all required Platform software has been installed (see Part III).

The Platform is the core transactional and business logic of OneSpan Sign.

The Platform is configured via the Administration Console, as described in the Platform Administrator’s Guide. Indeed, that entire guide contains information relevant to configuring the Platform.

By contrast, the following procedure describes how to perform just the minimum configuration needed to obtain a working version of the Platform.

To complete the following procedure, you must perform procedures described in the Platform Administrator’s Guide (e.g., Accessing the Web-based Admin Console, Creating/Editing/Deleting Configuration Properties, Managing Tenants, Creating/Updating/Deleting a Resource).

To quickly obtain a working version of the Platform, configure it as follows:

1. Configure the System Tenant by doing the following:
   a. Access the Admin Console, using Tenant = system.
   b. Under the Main Menu, click Manage Settings.
   c. Appropriately configure the properties in the following files, using as a reference Appendices B and C of the Platform Administrator’s Guide:
      - hibernate.properties – To find this file, click Global Properties > configurations.
      - apiSecurity.properties – To find this file, click Global Properties > configurations.
      - mailserverconfig.properties – To find this file, click e-SignLive Notifier > configurations.
      - silanis_quartz.properties – To find this file, click e-SignLive Scheduler > configurations.
      - documentEngine.properties – To find this file, click e-SignLive Core > configurations. Minimally, you must configure the parameter aws.document.engine.url to point to your cluster of PDF Document Engines.
      - tsapi.properties – To find this file, click e-SignLive TransactionStatus API > configurations. Minimally, you must configure the parameters database.server.type and database.schema.version. The value of the latter must be 5.0.
   d. If you plan to use e-Witness, appropriately configure the properties in the following files, using as a reference Appendix C of the Platform Administrator’s Guide:
      - reviewer.properties – To find this file, click e-SignLive e-Witness > configurations.
• keyStore.properties — Same location.
• printtopdf.properties — Same location.

e. Under the Main Menu, click Manage Tenants. On the page that appears, click Add to create a new tenant, giving it an appropriate name (e.g., esl).

The default name of the tenant used by the Application within the Platform is esl.

This tenant will inherit all the configurations of the previous steps.

f. Log out of the Admin Console.

2. Configure the new tenant by doing the following:

a. Access the Admin Console, specifying as Tenant the name you gave to the new tenant in Step 1e.

b. Under the Main Menu, click Manage Settings > e-SignLive Core > resources > aws-groovy-library > Add to add the file "eSL-Groovy-Controller.jar" provided by OneSpan. This file is the Application Controller.

c. Log out of the Admin Console.

3. If you are using RHEL/CentOS 6, the system might take a long time to create the first one or two transactions after a Platform server is recycled. To avoid this:

   a. Set the following JVM parameter:

   • -Djava.security.egd=file:/dev/urandom

   b. Install haveged (version 1.9.1-2.el6; architecture x86_64).

4. If you are using JBoss or Wildfly, edit the Platform file ...
   /standalone/configuration/esep.xml by adding the part of the following code that specifies max-post-size:

   ```
   <server name="default-server">
     <http-listener name="default" max-post-size="31457280" socket-binding="http"/>
   </server>
   ```

5. Ensure that the PDF Document Engine was properly configured in Chapter 11 and Appendix G.

   If you’re planning to use the REST API or the OneSpan Sign Application, before you try to use either, configure them using Chapter 18.
To further customize your configuration of the Platform, see the Platform Administrator's Guide.

This procedure ends here.
CHAPTER 18: Configuring the Application

The Application is essentially the front end that before OneSpan Sign 6.0 was available only via a SaaS deployment.

The Application is configured largely via OneSpan Sign BackOffice, as described in the Application Administrator’s Guide. Indeed, that entire guide contains information relevant to configuring the Application.

By contrast, the following procedure describes how to attain just the minimum configuration needed for a working version of the Application.

Prerequisites

• All required and desired Application components have been installed (see Chapter 15).
• You are the BackOffice Administrator for an on-premises deployment.
• If you want to use SMS, you must have an account with Twilio.
• If you want to use TLS 1.2 or Twilio SMS on WebSphere 8 or 9, you must configure the Application Server with the following generic Java Virtual Machine (JVM) argument:
  
  ```java
  -Dcom.ibm.jsse2.overrideDefaultTLS=true
  ```
• If you want to use OneSpan Sign Single Sign-On:
  a. On the Application Server host, an [SSO_HOME] directory has been created that will serve as the home directory for OneSpan Sign Single Sign-On.
  b. The SSO_HOME system variable has been defined to point to the [SSO_HOME] directory.

Default BackOffice Users

After BackOffice is installed and a relevant database is configured, the following two BackOffice Administrator accounts become active:

• The first account has the default username admin (default password = password). By default, it can only manage BackOffice users. However, it has Admin privileges, so it can grant itself additional rights.
• The second account has the default username settingsEditor (default password = password). It can manage resources for all accounts, but it can’t do anything else.

Action

To quickly obtain a working version of the Application, configure it as
Configuring the Application

follows:

1. Navigate to the URL of your BackOffice installation (since that URL is not exposed to the public, you must be inside your network). A Login screen appears.

2. Log in to BackOffice using the username *settingsEditor* and its associated password. A Welcome screen appears.

3. Click **Manage Resources**. The Resources for all Accounts screen appears.

4. Edit the parameters in Table 18-1, and optionally edit the parameters in Table 18-2. The associated Resource Keys are listed in the leftmost column (e.g., *settings.properties*). To edit the parameters for a particular key, click the *Edit* icon in its rightmost column. A new screen displays the key’s parameters. When you've finished editing those parameters, click *Save*.

   **Multiple User Accounts in the Application can be mapped to one or more Platform tenants** (you cannot map one Account to multiple tenants). Two examples: (1) all Accounts are mapped to one tenant; (2) most Accounts are mapped to one tenant, while all others are mapped individually to dedicated tenants. This architecture provides the flexibility to customize your deployment strategy according to your performance, maintenance, and logistical needs. **To achieve the desired mapping, configure appropriate multiplicity for each of Table 18-1’s Resource Keys.** Note that while there can be only one Account per *esl-core.properties*’ *tenant*, there can be multiple such pairs. If you want more than one pair, you must create multiple copies of *esl-core.properties* (i.e., multiple database rows), one for each pair. Similar remarks apply to the pairing of an Account with each of the following parameters: (1) *tsapi.config*’s *context_id*; (2) *email.properties*’ *email.smtp.host*; (3) *email.properties*’ *email.smtp.port*; (4) *settings.properties*’ *esl.url.uiresources*; (5) *logout.config*’s *esl.url.uiresources/login*.

5. If you want to use OneSpan Sign Single Sign-On:
   a. Create a file called *saml.properties*. In it, appropriately specify the parameters in Table 18-3.

   **Security Assertion Markup Language** (SAML) is an XML-based, open-standard data format for exchanging authentication and authorization data between parties.
b. Create a keystore for SAML called samlKeystore.jks. To create a keystore for a production environment, see Appendix E.

c. Copy the files saml.properties and samlKeystore.jks into the [SSO_HOME] directory.

d. Configure SAML for all relevant user accounts (for detailed instructions, see Configuring SAML for User Accounts in the Application Administrator’s Guide).

6. To enable reCAPTCHA functionality:
   a. Sign up for an API key pair using the following URL:

   https://developers.google.com/recaptcha/intro

   b. In the sender-ui directory on the server where the OneSpan Sign Application Frontend is installed, open the file settings.js. Typically, that file is in the following location:

   /opt/esignlive/frontend-sender-ui/frontend/settings.js

   c. Configure the following settings in that file:

   ```
   RECAPTCHA_ENABLED: env.bool('ESL_ENABLE_RECAPTCHA', true),
   RECAPTCHA_SITE_KEY: env(
       'RECAPTCHA_SITE_KEY',
       '<your site key>'
   ),
   RECAPTCHA_SECRET_KEY: env(
       'RECAPTCHA_SECRET_KEY',
       // Test key retrieved from
       // https://developers.google.com/recaptcha/docs/faq
       // '6LeIxAcTAAAAAGG-vFI1TnRxMzNPsuoJ4WifJWe'
       '<your secret key>'
   )
   ```

   d. Log in to OneSpan Sign BackOffice. For the account on which you’d like to enable reCAPTCHA, update settings.properties with the following settings:

   ```
   "esl.recaptcha.url" :
   "https://www.google.com/recaptcha/api/siteverify"
   "esl.recaptcha.secret" : "<your secret key>"
   ```

7. If you are using JBoss or Wildfly, edit the Application file /standalone/configuration/esep.xml by adding the part of the following code that twice specifies `max-post-size`:

   ```xml
   <subsystem xmlns="urn:jboss:domain:undertow:1.2">
     <buffer-cache name="default"/>
     <server name="default-server">
       <http-listener name="default" max-post-size="26214400" socket-binding="http"/>
     </server>
   </subsystem>
   ```
8. Restart the Application Server.

To further customize your configuration of the Application, see the Application Administrator’s Guide.

This procedure ends here.

**Table 18-1: Configuration parameters required by the Application**

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Resource Key = esl-core.properties</strong></td>
<td>This key configures the connection to the Core Component. Required Format:</td>
</tr>
<tr>
<td>{ &quot;baseURL&quot; : &quot;$baseURL&quot;, &quot;tenant&quot; : &quot;$tenant&quot;, &quot;script&quot; : &quot;$script&quot; }</td>
<td>Required Parameters:</td>
</tr>
<tr>
<td>• $baseURL is a placeholder for the base URL of the Core Component, followed immediately by: /esep/rest/api/v1/script/application/%s/execute/%s</td>
<td></td>
</tr>
<tr>
<td>• $tenant is a placeholder for the name of the tenant (default = esl)</td>
<td></td>
</tr>
<tr>
<td>• $script is a placeholder for the name of the required Groovy script. The value of this parameter must be: com.silanis.workflow.controller.Dispatcher.groovy</td>
<td></td>
</tr>
<tr>
<td><strong>Resource Key = tsapi.config</strong></td>
<td>This key configures the Transaction Status API. Required Format:</td>
</tr>
<tr>
<td>{ &quot;context_id&quot; : &quot;$context_id&quot;, &quot;base_service_endpoint&quot; : &quot;$base_service_endpoint&quot;, &quot;service_endpoint&quot; : &quot;$service_endpoint&quot;, &quot;username&quot; : &quot;&quot;, &quot;token&quot; : &quot;&quot; }</td>
<td>Required Parameters:</td>
</tr>
<tr>
<td>• $context_id is a placeholder for the name of the tenant.</td>
<td></td>
</tr>
<tr>
<td>• $base_service_endpoint is a placeholder for the Core Component’s base URL used to load a transaction’s signers, followed immediately by: /tsapi/services/rest/v3</td>
<td></td>
</tr>
<tr>
<td>• $service_endpoint is a placeholder for the Core Component’s base URL used to load transactions, followed immediately by: /tsapi/esl/services/rest/v3/transactions/</td>
<td></td>
</tr>
</tbody>
</table>
Configuring the Application

Resource Key = email.properties

This key configures the SMTP e-mail server. **Note:** E-mail services must be configured either via a generic SMTP server or via Amazon SES (*Simple Email Service*). However, only the latter offers bounce-back functionality (i.e., the sender of an invitation will receive an alert if the invitation is bounced from a recipient’s account).

Required Format:

```json
{ "email.smtp.host" : "$email.smtp.host", "email.smtp.port" : "$email.smtp.port",
  "email.username" : "$email.username", "email.password" : "$email.password",
  "email.smtp.ssl" : "$email.smtp.ssl", "email.smtp.tls" : "$email.smtp.tls" }
```

Required Parameters:

- `$email.smtp.host` is a placeholder for the host name of the SMTP server.
- `$email.smtp.port` is a placeholder for the port number of the SMTP server.
- `$email.username` is a placeholder for the username used with the SMTP server.
- `$email.password` is a placeholder for the password used with the SMTP server.
- `$email.smtp.ssl` must have the value `false` (= off).
- `$email.smtp.tls` can be `true` or `false` (TLS security on/off).
Resource Key = settings.properties

This key configures miscellaneous Application properties.

Required Format:

```json
{ "aws.core.content_encryption.enable" : "false", "aws.core.counters.enabled" : "true", "aws.document.autofield.date.format" : "yyyy-MM-dd HH:mm:ss 'GMT'",
  "esl.url.uiresources" : "$esl.url.uiresources",
  "esl.url.account.invitation" : "$esl.url.account.invitation", "esl.url.forgot_password" : "$esl.url.forgot_password",
  "esl.login.maxFailedAttempts" : "10",
  "esl.login.session.timeout" : "1800000", "esl.url.fastTrack" : "$esl.url.fastTrack/ft?token=%s",
  "esl.url.shortener.host" : "$esl.url.shortener.host",
  "esl.url.shortener.public.host" : "$esl.url.shortener.public.host",
  "esl.package.list.expiringSoonDays" : 5,
  "esl.email.maxAttachmentSize" : "5242880",
  "esl.authenticationToken.expirationDelay.minutes" : "30",
  "esl.signerAuthenticationToken.expirationDelay.minutes" : "30",
  "esl.authenticationCode.expirationDelay.minutes" : 5 }
```

Required Parameters:

- `$esl.url.uiresources` is a placeholder for the OneSpan Sign Application Frontend’s publicly visible URL (e.g., https://esigning.company.com/ — including the final forward slash).
- `$esl.url.account.invitation` is a placeholder for the OneSpan Sign Application Frontend’s publicly visible URL, followed immediately by: /signup
- `$esl.support.email` is a placeholder for the support e-mail address that will be displayed to users.
- `$esl.support.phoneNumber` is a placeholder for the support phone number that will be displayed to users.
Resource Key = sms.provider

This key configures Twilio SMS.

Required Format:

```json
{
  "esl.notification.sms.provider" : "TWILIO",
  "esl.notification.sms.twilio.sid" : "$esl.notification.sms.twilio.sid",
  "esl.notification.sms.twilio.number" : "$esl.notification.sms.twilio.number",
  "esl.notification.sms.twilio.auth.token" : "$esl.notification.sms.twilio.auth.token",
  "esl.notification.sms.twilio.endpoint" : "https://api.twilio.com",
}
```

Required Parameters:

- `$esl.notification.sms.twilio.sid` is a placeholder for the SID of your Twilio application.
- `$esl.notification.sms.twilio.number` is a placeholder for the number from which SMS messages will appear to be sent.
- `$esl.notification.sms.twilio.auth.token` is a placeholder for the Auth Token used by Twilio.

Resource Key = converter.settings

This key configures the `OneSpan Sign Document Converter`.

Required Format:

```json
{
  "esl.converter.aspose" : {
    "primary" : "$esl.converter.aspose",
    "fallback" : ""
  }
}
```

Required Parameters:

- `$esl.converter.aspose` is a placeholder for the URL of the Aspose `OneSpan Sign Document Converter` (e.g., "http://asposeconverter/OpenOfficeConverter/pdf").
Configuring the Application

**Resource Key = logout.config**

This key configures where users are redirected when either they log out or their session expires.

**Required Format:**

```
{  
  "senderExpiryRedirect" : "$esl.url.uiresources/login",
  "logoutRedirect" : "$esl.url.uiresources/login",
  "signerExpiryRedirect" : "$esl.url.uiresources/login"
}
```

**Required Parameters:**

- `$esl.url.uiresources/login` is a placeholder for the OneSpan Sign Application Frontend’s publicly visible URL, followed immediately by: /login

**Resource Key = encryption.properties**

This key configures the parameters required to encrypt sensitive fields in the OneSpan Sign Application Database.

**Required Format:**

```
{
  "encryption.password" : "$encryption.password",
  "encryption.salt" : "$encryption.salt"
}
```

**Required Parameter:**

- `$encryption.password` is a placeholder for the password used to perform the encryption. You must update this parameter, but you can do it only once. After the parameter is updated, the system will encode the password and generate a random salt (do not try to specify $encryption.salt). At that point, the encryption password will become read-only. The password you specify must have at least 8 characters, and satisfy at least three of the following conditions: (1) there is one or more uppercase characters; (2) there is one or more lowercase characters; (3) there is one or more special characters (~ ! @ $ % ^ & *); (4) there is one or more digits (0-9).
### Table 18-2: Optional configuration parameters for the Application

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Resource Key</strong> = signer.verification.encryption.properties**</td>
<td>This key configures the parameters required to encrypt sensitive fields in the payload for the <em>External Signer Verification</em> feature.</td>
</tr>
<tr>
<td>Required Format:</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>&quot;encryption.password&quot; : &quot;$encryption.password&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;encryption.salt&quot; : &quot;$encryption.salt&quot;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td><strong>Required Parameter:</strong></td>
<td></td>
</tr>
<tr>
<td>$encryption.password** is a placeholder for the password used to perform the encryption. You must update this parameter, but you can do it only once. After the parameter is updated, the system will encode the password and generate a random salt (do not try to specify encryption.salt). At that point, the encryption password will become read-only. The password you specify must have at least 8 characters, and satisfy at least three of the following conditions: (1) there is one or more uppercase characters; (2) there is one or more lowercase characters; (3) there is one or more special characters (~ ! @ # $ % ^ &amp; *); (4) there is one or more digits (0-9).</td>
<td></td>
</tr>
<tr>
<td><strong>Resource Key</strong> = esignliveLogo**</td>
<td>This key is the file <em>esignLiveLogo</em>, which contains the eSignLive logo. It can be found in the directory /common/blobs of the unzipped content that you extracted in performing the Appendix A procedure <em>Creating the Application Database</em>.</td>
</tr>
<tr>
<td><strong>Resource Key</strong> = default.consent.document**</td>
<td>This key is the file <em>default.consent.document</em>, which is the default consent document displayed to users. It can be found in the directory /common/blobs of the unzipped content that you extracted in performing the Appendix A procedure <em>Creating the Application Database</em>.</td>
</tr>
<tr>
<td><strong>Resource Key</strong> = email.blacklist**</td>
<td>This key specifies email domains that will never receive e-mail from the <em>OneSpan Sign Application</em>.</td>
</tr>
<tr>
<td>Required Format:</td>
<td></td>
</tr>
<tr>
<td>[]</td>
<td></td>
</tr>
<tr>
<td><strong>Values:</strong></td>
<td></td>
</tr>
<tr>
<td>• The domain names must be in a comma-separated list. For example: [&quot;gmail.com&quot;, &quot;outlook.com&quot;]</td>
<td></td>
</tr>
</tbody>
</table>
Resource Key = kba.equifax.usa.config

This key configures the Equifax USA service.

Required Format:

{ "maxAttempts" : "2", "username" : "$username", "token" : "$token" }

Optional Parameters:

• "maxAttempts" is the maximum number of KBA attempts. Valid values are 1 and 2. There is no default value.
• $username is a placeholder for your Equifax USA username.
• $token is a placeholder for your Equifax USA security token.

Resource Key = kba.equifax.canada.config

This key configures the Equifax Canada service.

Required Format:

{ "maxAttempts" : "2", "username" : "$username", "token" : "$token" }

Optional Parameters:

• "maxAttempts" is the maximum number of KBA attempts. Valid values are 1 and 2. There is no default value.
• $username is a placeholder for your Equifax Canada username.
• $token is a placeholder for your Equifax Canada security token.
Resource Key = settings.properties

This key configures miscellaneous Application properties.

Required Format:

```json
```

Optional Parameters:

- `$esl.url.qrcode` is a placeholder for the OneSpan Sign Application Frontend’s publicly visible URL, followed immediately by: /qr/$guid;/$document;
- `$esl.url.forgot_password` is a placeholder for the OneSpan Sign Application Frontend’s publicly visible URL, followed immediately by: /resetpassword
- `$esl.url.fastTrack` is a placeholder for the OneSpan Sign Application Frontend’s publicly visible URL, followed immediately by: /ft?token=%s
- `$esl.url.shortener.host` is a placeholder for the OneSpan Sign Application Backend’s URL, followed immediately by: /UrlMapper
- `$esl.url.shortener.public.host` is a placeholder for the OneSpan Sign Application Backend’s publicly visible URL, followed immediately by: /UrlMapper
- `$esl.url/equifax-service` is a placeholder for the OneSpan Sign Application Backend’s URL, followed immediately by: /equifax-service
### Table 18-3: Configuration parameters in "saml.properties"

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>saml.metadata.baseurl</td>
<td>URL of the SSO Web service</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Required format:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>https://<a href="">server:port</a>/sso</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="https://apps.eSignLive.com/sso">https://apps.eSignLive.com/sso</a></td>
</tr>
<tr>
<td>saml.metadata.entityid</td>
<td>Entity ID of the SSO service provider</td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>urn:saml:sso:apps: eSignLive:com:</td>
</tr>
<tr>
<td>saml.metadata.entity.alias</td>
<td>Entity Alias of the SSO service provider</td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>eSignLive</td>
</tr>
<tr>
<td>saml.admin.username</td>
<td>Username of the SSO administrator</td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>saml-admin</td>
</tr>
<tr>
<td>saml.admin.password</td>
<td>Encrypted password of the SSO administrator</td>
<td>The creation of this encrypted password is similar to that of CPSPassword in Table 11-2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Required format:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>http://<a href="">server:port</a>/aws/auth/saml/preauthenticated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="http://localhost:7001/aws/auth/saml/preauthenticated">http://localhost:7001/aws/auth/saml/preauthenticated</a></td>
</tr>
<tr>
<td>serviceprovider.restapi.</td>
<td>The REST API to which the SAML server sends user</td>
<td>Required format:</td>
</tr>
<tr>
<td>preauthenticated</td>
<td>information, and then receives a redirect URL</td>
<td>http://<a href="">server:port</a>/aws/auth/saml</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="http://localhost:8080/aws/auth/saml">http://localhost:8080/aws/auth/saml</a></td>
</tr>
<tr>
<td>serviceprovider.restapi.idplist</td>
<td>The REST API from which the SAML server receives a list of identity providers</td>
<td>Required format:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/aws/auth/saml/metadata</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="http://localhost:8080/aws/auth/saml/metadata">http://localhost:8080/aws/auth/saml/metadata</a></td>
</tr>
<tr>
<td>PARAMETER</td>
<td>DESCRIPTION</td>
<td>VALUE</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>serviceprovider.</td>
<td>Endpoint of the failure of an SSO</td>
<td>Required format:</td>
</tr>
<tr>
<td>restapi.login.error</td>
<td>authentication process</td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>https://&lt;server&gt;/login?errormsg=</td>
</tr>
<tr>
<td>jks.keystore.file</td>
<td>Path of the SAML keystore</td>
<td>Default: ${SSO_HOME}/samlKeystore.jks</td>
</tr>
<tr>
<td>jks.keystore.</td>
<td>Encrypted password of the SAML</td>
<td>Default: 2.1000.SVDegZllzi4= .baZeExmoRmkyXO6t kvbgFQ==</td>
</tr>
<tr>
<td>password</td>
<td>keystore</td>
<td></td>
</tr>
<tr>
<td>jks.keystore.alias</td>
<td>Alias of the certificate in the keystore used by the</td>
<td>Default: e-SignLive</td>
</tr>
<tr>
<td></td>
<td>SSO service provider</td>
<td></td>
</tr>
<tr>
<td>jks.keystore.alias.</td>
<td>Encrypted password of the alias of the certificate</td>
<td>Default: 2.1000.SVDegZllzi4= .baZeExmoRmkyXO6t kvbgFQ==</td>
</tr>
<tr>
<td>password</td>
<td>in the keystore used by the SSO service provider</td>
<td></td>
</tr>
<tr>
<td>idp.entityid.default</td>
<td>Default Entity ID of the identity provider</td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="https://openidp.feide.no">https://openidp.feide.no</a></td>
</tr>
<tr>
<td>idp.metadata.url.default</td>
<td>Default URL of the identity provider's metadata</td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="https://openidp.feide.no/simplesaml/saml2/idp/metadata.php">https://openidp.feide.no/simplesaml/saml2/idp/metadata.php</a></td>
</tr>
<tr>
<td>key.user.email</td>
<td>SAML Authentication attributes used to extract</td>
<td>Default (a comma-separated list of attributes):</td>
</tr>
<tr>
<td></td>
<td>information about a user’s e-mail</td>
<td>email,emailaddress, mail</td>
</tr>
<tr>
<td>key.userfirstname</td>
<td>SAML Authentication attributes used to extract</td>
<td>Default (a comma-separated list of attributes):</td>
</tr>
<tr>
<td></td>
<td>information about a user’s first name</td>
<td>firstname,givenname, cn</td>
</tr>
<tr>
<td>PARAMETER</td>
<td>DESCRIPTION</td>
<td>VALUE</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>key.user.lastname</td>
<td>SAML Authentication attributes used to extract information about a user’s last name</td>
<td>Default (a comma-separated list of attributes): lastname,surname,sn</td>
</tr>
<tr>
<td>proxy.host</td>
<td>Proxy Server used when the SAML server makes calls to external services</td>
<td>Default: &lt;empty&gt;</td>
</tr>
<tr>
<td>proxy.port</td>
<td>Port of the Proxy Server used when the SAML server makes calls to external services</td>
<td>Default: &lt;empty&gt;</td>
</tr>
<tr>
<td>proxy.excluded.hosts</td>
<td>List of hosts of services that can be called directly by the SAML server without going through a Proxy Server</td>
<td>Default: &lt;empty&gt;</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This parameter is relevant only if you have configured proxy.host and proxy.port.</td>
<td>Example: localhost</td>
</tr>
</tbody>
</table>
# Part VIII: Appendices

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: Creating Databases</td>
<td>280</td>
</tr>
<tr>
<td>B: Configuring Logging</td>
<td>291</td>
</tr>
<tr>
<td>C: Configuring Cryptographic Libraries</td>
<td>299</td>
</tr>
<tr>
<td>D: Expected Log-File Contents</td>
<td>300</td>
</tr>
<tr>
<td>E: Creating a Keystore</td>
<td>306</td>
</tr>
<tr>
<td>F: FIPS 140-2 Compliance</td>
<td>308</td>
</tr>
<tr>
<td>G: Tenant Environments</td>
<td>309</td>
</tr>
<tr>
<td>H: App Server Procedures</td>
<td>312</td>
</tr>
<tr>
<td>I: Configuring IIS &amp; Optimizing Performance</td>
<td>319</td>
</tr>
<tr>
<td>J: Running the DE as an Application User</td>
<td>322</td>
</tr>
<tr>
<td>K: Glossary</td>
<td>327</td>
</tr>
</tbody>
</table>
APPENDIX A: Creating Databases

Several chapters in this guide ask you to create one or more of the following databases:

- **Core Database** — A database used by the Core Component
- **e-Mail Notification Manager Database** — A database used by the e-Mail Notification Manager™
- **Archive Database** — A database used by the Archival Module to archive a Production Database
- **Audit Database** — A database used by the Audit Service
- **User Manager Database** — A database used by the User Manager
- **OneSpan Sign Application Database** — A database used by the OneSpan Sign Application

The following sections describe how to create those databases:

- Creating the Core Database (page 280)
- Creating the e-Mail Notification Manager Database (page 283)
- Creating the Archive Database (page 284)
- Creating the Audit Database (page 287)
- Creating the User Manager Database (page 287)
- Creating the Application Database (page 288)

The procedures in this appendix assume that your environment meets all relevant requirements in Chapter 2.

Except for the Archive Database, all database tables in this appendix can be stored either in the Core Database or in a separate database.

Creating the Core Database

The Core Component requires the Core Database. This section describes how to create that database.

If you are in a tenant environment, and want a separate Core Database for each tenant, you must create a separate schema and Data Source for each tenant. This "multi-database" configuration applies only to the Core Database, not to any other database in this appendix.

Directory and file names in this section often include variables of the form \[X \text{ or } Y\]. You can find each script in this section in one of the following directories (see the Packaging section of Chapter 2): "Server/Core Components/Database/Core DDL/[default or partitioned]/[all or english]".

Prerequisites

- The following scripts from the software provided by OneSpan have been
copied to your hard drive:

- If you use a DB2 database, and if you want partitioning (recommended), copy the scripts "db2_tablespaces.sql" and "db2_schema_tbsp.sql" from the directory "Server/Core Components/Database/Core DDL/partitioned/[all or english]/db2". These scripts are used in Step 2 below.
- If you use a DB2 database and you don’t want partitioning, copy the scripts "db2_tablespaces.sql" and "db2_schema_tbsp.sql" from the directory "Server/Core Components/Database/Core DDL/default/[all or english]". These scripts are used in Step 3 below.
- If you use an Oracle database, and if you want partitioning (recommended), copy the scripts "oracle_tablespaces.sql" and "oracle_schema_tbsp.sql" from the directory "Server/Core Components/Database/Core DDL/partitioned/[all or english]/oracle". These scripts are used in Step 4 below.
- If you use an Oracle database and you don’t want partitioning, copy the scripts "oracle_tablespaces.sql" and "oracle_schema_tbsp.sql" from the directory "Server/Core Components/Database/Core DDL/default/[all or english]". These scripts are used in Step 5 below.
- If you use an SQL Server database, copy the scripts "sqlserver_tablespaces.sql" and "sqlserver_schema_tbsp.sql" from the directory "Server/Core Components/Database/Core DDL/default/[all or english]". These scripts are used in Step 6 below.
- If you use a MySQL database, copy the scripts "mysql.sql" from the directory "Server/Core Components/Database/Core DDL/default/[all or english]". This script is used in Step 7 below.
- If you use a MySQL 5.7 database, open the MySQL configuration file, and specify the following setting:
  
  sql_mode=ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION

- If a Core Database already exists, if you want to create an additional one, and if you plan to use the Electronic Evidence™ Export Utility, ensure that there are no contents in the directory where that utility’s PDF files and Web Archive HTML files are stored (see the pdfPath parameter in Table C-10 of the Platform Administrator’s Guide).

**Action**

**To create the Core Database and its tables:**

1. Add a new database to the Core Database Server, and assign it a meaningful name (e.g., "esep"). If you are adding a DB2 database, set the page size of the default buffer pool to 32768 bytes. If you are adding a MySQL database, set the charset to UCS-2.

2. If you added a DB2 database in Step 1, and you want partitioning:
   a. Create a user who will be able to access that database.
   b. Create tablespaces for that database by running a script based on the sample script "db2_tablespaces.sql" (you will likely want to customize that script).
c. Create tables, indexes, etc. for that database by running the script "db2_schema_tbsp.sql".

⚠️ You must configure the system to set the connecting user’s schema when a DB2 database connection is made. For example, you could configure the Application Server’s Data Source to execute the statement "SET SCHEMA DB2INST1" upon creating new connections.

3. If you added a DB2 database in Step 1, and you do not want partitioning:
   a. Create tablespaces for that database by running a script based on the sample script "db2_tablespaces.sql" (you will likely want to customize that script).
   b. Create tables, indexes, etc. for that database by running the script "db2_schema_tbsp.sql".

   ℹ️ Step 3 does not create a specific user to access the database, or grant user permissions. If you use Step 3, you may wish to do these things manually. Step 2 does these things (as well as creating tables, etc.).

4. If you added an Oracle database in Step 1, and you want partitioning:
   a. Create a user who will be able to access that database.
   b. Create tablespaces for that database by running a script based on the sample script "oracle_tablespaces.sql" (you will likely want to customize that script).
   c. Create tables, indexes, etc. for that database by running the script "oracle_schema_tbsp.sql".

   ⚠️ You must configure the system to set the connecting user’s schema when an Oracle database connection is made. For example, you could configure the Application Server’s datasource to execute the statement "SET SCHEMA ORACLEINST1" upon creating new connections.

5. If you added an Oracle database in Step 1, and you do not want partitioning:
   a. Create tablespaces for that database by running a script based on the sample script "oracle_tablespaces.sql" (you will likely want to customize that script).
   b. Create tables, indexes, etc. for that database by running the script "oracle_schema_tbsp.sql".

   ℹ️ Step 5 does not create a specific user to access the database, or grant user permissions. If you use Step 5, you may wish to do these things manually. Step 4 does these things (as well as creating tables, etc.).

6. If you added an SQL Server database in Step 1:
   a. Create tablespaces for that database as follows: (1) open the script "sqlserver_tablespaces.sql"; (2) specify the parameter
Creating the e-Mail Notification Manager Database

The e-Mail Notification ManagerTM requires the e-Mail Notification ManagerTM Database. This section describes how to create that database.

The database tables for the e-Mail Notification ManagerTM can be stored either in the Core Database or in a separate database. The following procedure permits you to use either option.

Prerequisites

- The following script from the software provided by OneSpan (in the directory "Server/Core Components/Required Modules/e-Mail Notification Manager/Database") has been copied to your hard drive:
  - For a DB2 database, the script = Scheduler_db2_schema.sql
  - For an SQL Server database, the script = Scheduler_mssql_schema.sql
  - For an Oracle database, the script = Scheduler_oracle_schema.sql
  - For a MySQL database, the script = Scheduler_mysql_schema.sql

Action

To create the e-Mail Notification ManagerTM Database:

1. If you want the database tables for the e-Mail Notification ManagerTM to be in a separate database, create an empty database, and assign it a meaningful name (e.g., "notifmanager_db"). If you are adding a DB2 database, set the page size of the default buffer pool to 32768 bytes. If
you are adding a MySQL database, set the \textit{charset} to UCS-2.

2. Run the following script located in the software provided by OneSpan, targeting either the Core Database or the database you created in \textbf{Step 1}:
   - For a DB2 database, \texttt{script = Scheduler\_db2\_schema.sql}
   - For an SQL Server database, \texttt{script = Scheduler\_mssql\_schema.sql}
   - For an Oracle database, \texttt{script = Scheduler\_oracle\_schema.sql}
   - For a MySQL database, \texttt{script = Scheduler\_mysql\_schema.sql}

3. Verify that the script executed successfully.
   
   This procedure ends here.

\section*{Creating the Archive Database}

The \textit{Archival Module} requires the Archive Database. This section describes how to create that database.

\begin{itemize}
\item If you are using the "Multi-Database Model" for the Core Database, you must create a separate schema and Data Source for each tenant.
\end{itemize}

\textbf{Prerequisites}

\begin{itemize}
\item The following scripts from the software provided by OneSpan have been copied to your hard drive:
  \begin{itemize}
  \item If you use a DB2 database, and if you want partitioning (recommended), copy the scripts "db2\_tablespaces.sql" and "db2\_schema\_tbsp.sql" from the directory "Server/Core Components/Database/Core DDL/partitioned/[all or english]/db2". These scripts are used in \textbf{Step 2} below.
  \item If you use a DB2 database and you don't want partitioning, copy the scripts "db2\_tablespaces.sql" and "db2\_schema\_tbsp.sql" from the directory "Server/Core Components/Database/Core DDL/default/[all or english]/db2". These scripts are used in \textbf{Step 3} below.
  \item If you use an Oracle database, and if you want partitioning (recommended), copy the scripts "oracle\_tablespaces.sql" and "oracle\_schema\_tbsp.sql" from the directory "Server/Core Components/Database/Core DDL/partitioned/[all or english]/oracle". These scripts are used in \textbf{Step 4} below.
  \item If you use an Oracle database and you don't want partitioning, copy the scripts "oracle\_tablespaces.sql" and "oracle\_schema\_tbsp.sql" from the directory "Server/Core Components/Database/Core DDL/default/[all or english]/oracle". These scripts are used in \textbf{Step 5} below.
\end{itemize}

Directory and file names in this section often include variables of the form \[X\text{ or }Y\]. You can find each script in this section in one of the following directories (see the \textit{Packaging} section of Chapter 2): "Server/Core Components/Database/Core DDL/[default or partitioned]/[all or english]".
• If you use an SQL Server database, copy the scripts "sqlserver_tablespaces.sql" and "sqlserver_schema_tbsp.sql" from the directory "Server/Core Components/Database/Core DDL/default/[all or english]". These scripts are used in Step 6 below.

• If you use a MySQL database, copy the script "mysql.sql" from the directory "Server/Core Components/Database/Core DDL/default/[all or english]". This script is used in Step 7 below.

Note that the scripts in the above bullets are — and must be — identical to those used in the procedure Creating the Core Database.

• The script "archive_final_doc.sql" from the directory "Server/Optional Modules/Archival Module/Database/[db2 or oracle or sqlserver or mysql], where the last sub-directory depends on your database type. This script is used in Step 9 below.

Action

To create the Archive Database:

1. Add a new database to the Archival Module’s Database Server, and assign it a meaningful name (e.g., "archive_ese"). If you are adding a DB2 database, set the page size of the default buffer pool to 32768 bytes. If you are adding a MySQL database, set the charset to UCS-2.

2. If you added a DB2 database in Step 1, and you want partitioning:
   a. Create a user who will be able to access that database.
   b. Create tablespaces for that database by running a script based on the sample script "db2_tablespaces.sql" (you will likely want to customize that script).
   c. Create tables, indexes, etc. for that database by running the script "db2_schema_tbsp.sql".

You must configure the system to set the connecting user’s schema when a DB2 database connection is made. For example, you could configure the Application Server’s Data Source to execute the statement "SET SCHEMA DB2INST1" upon creating new connections.

3. If you added a DB2 database in Step 1, and you do not want partitioning:
   a. Create tablespaces for that database by running a script based on the sample script "db2_tablespaces.sql" (you will likely want to customize that script).
   b. Create tables, indexes, etc. for that database by running the script "db2_schema_tbsp.sql".

Step 3 does not create a specific user to access the database, or grant user permissions. If you use Step 3, you may wish to do these things manually. Step 2 does these things (as well as creating tables, etc.).
4. If you added an Oracle database in Step 1, and you want partitioning:
   a. Create a user who will be able to access that database.
   b. Create tablespaces for that database by running a script based on the sample script "oracle_tablespaces.sql" (you will likely want to customize that script).
   c. Create tables, indexes, etc. for that database by running the script "oracle_schema_tbsp.sql".

   ![Warning]
   You must configure the system to set the connecting user's schema when an Oracle database connection is made. For example, you could configure the Application Server's datasource to execute the statement "SET SCHEMA ORACLEINST1" upon creating new connections.

5. If you added an Oracle database in Step 1, and you do not want partitioning:
   a. Create tablespaces for that database by running a script based on the sample script "oracle_tablespaces.sql" (you will likely want to customize that script).
   b. Create tables, indexes, etc. for that database by running the script "oracle_schema_tbsp.sql".

   ![Note]
   Step 5 does not create a specific user to access the database, or grant user permissions. If you use Step 5, you may wish to do these things manually. Step 4 does these things (as well as creating tables, etc.).

6. If you added an SQL Server database in Step 1:
   a. Create tablespaces for that database as follows: (1) open the script "sqlserver_tablespaces.sql"; (2) specify the parameter <real_database_name>; (3) specify a data path; (4) save and then run the script.
   b. Create tables, indexes, etc. for that database by running the script "sqlserver_schema_tbsp.sql".

7. If you added a MySQL database in Step 1, create tables, indexes, etc. for that database by running the script "mysql.sql".

8. Verify that the script(s) in Step 2, Step 3, Step 4, Step 5, Step 6, or Step 7 executed successfully.

9. Run the script "archive_final_doc.sql". This creates a table that will hold the signed documents on the Archive Database. Verify that the script ran successfully.

10. If your Database Server is an Oracle Database with RAC, run the following SQL statement:

    ALTER SEQUENCE Seq.AWS_PAGE_LOG CACHE 1000 ORDER;

11. If the Core Database’s username and schema name are different, reconfigure the custom property "Current Schema" described in Table 7-1.

This procedure ends here.
Creating the Audit Database

The Audit Service of OneSpan Sign requires the Audit Database. This section describes how to create that database.

The database tables for the Audit Service can be stored either in the Core Database or in a separate database. The following procedure permits you to use either option.

Prerequisites

- The following script from the software provided by OneSpan (in the directory "Server/Core Components/Database/Audit DDL/english") has been copied to your hard drive:
  - For a DB2 database, the script = db2.sql
  - For an SQL Server database, the script = sqlserver.sql
  - For an Oracle database, the script = oracle.sql
  - For a MySQL database, the script = mysql.sql

Action

To create the Audit Database:

1. If you want the database tables for the Audit Database to be in a separate database, create an empty database, and assign it a meaningful name (e.g., "audit_db"). If you are adding a DB2 database, set the page size of the default buffer pool to 32768 bytes. If you are adding a MySQL database, set the charset to UCS-2.

2. Run the following script located in the software provided by OneSpan, targeting either the Core Database or the database you created in Step 1:
   - For a DB2 database, script = db2.sql
   - For an SQL Server database, script = sqlserver.sql
   - For an Oracle database, script = oracle.sql
   - For a MySQL database, script = mysql.sql

3. Verify that the script executed successfully.

This procedure ends here.

Creating the User Manager Database

The User Manager requires the User Manager Database. This section describes how to create that database.

The database tables for the User Manager can be stored either in the Core Database or in a separate database. The following procedure permits you to use either option.
**Prerequisites**

- The following script from the software provided by OneSpan (in the directory "\Server\Core Components\Required Modules\User Manager\Database\default\english") has been copied to your hard drive:
  - For a DB2 database, the script = db2.sql
  - For an SQL Server database, the script = sqlserver.sql
  - For an Oracle database, the script = oracle.sql
  - For a MySQL database, the script = mysql.sql

**Action**

**To create the User Manager Database:**

1. If you want the database tables for the User Manager to be in a separate database, create an empty database, and assign it a meaningful name (e.g., "usermanager_db"). If you are adding a DB2 database, set the page size of the default buffer pool to 32768 bytes. If you are adding a MySQL database, set the charset to UCS-2.

2. Run the following script located in the software provided by OneSpan, targeting either the Core Database or the database you created in Step 1:
   - For a DB2 database, script = db2.sql
   - For an SQL Server database, script = sqlserver.sql
   - For an Oracle database, script = oracle.sql
   - For a MySQL database, script = mysql.sql

3. Verify that the script executed successfully.

This procedure ends here.

**Creating the Application Database**

The OneSpan Sign Application requires the OneSpan Sign Application Database. This section describes how to create that database.

The database tables for the OneSpan Sign Application can be stored either in the Core Database or in a separate database. The following procedure permits you to use either option.

**Prerequisites**

- The database charset configuration must support the character sets of all languages you want to use.
- The following zip file from the software provided by OneSpan (in the directory "\Server\eSL Backend Components\Database") has been copied to your hard drive:
  - For a DB2 database, the file 16.28.11-db2.zip
For an SQL Server database, the file 16.28.11-mssql.zip
For an Oracle database, the file 16.28.11-oracle.zip
For a MySQL database, the file 16.28.11-mysql.zip

- Your JAVA_HOME has been configured correctly.
- If you are using WebSphere and an Oracle database, run the following commands as user SYS:

  grant select on pending_trans$ to <user>;
  grant select on dba_2pc_pending to <user>;
  grant select on dba_pending_transactions to <user>;
  grant execute on dbms_xa to <user>;

Here <user> is the username configured in the authentication alias of Oracle’s Data Source.

**Action**

**To create the OneSpan Sign Application Database:**

1. If you want the database tables for the OneSpan Sign Application to be in a separate database, create an empty database, and assign it a meaningful name (e.g., "esea_db"). If you are adding a DB2 database, set the page size of the default buffer pool to 32768 bytes. If you are adding a MySQL database, set the charset to UCS-2.

   We strongly recommend that you store the database tables in a separate database.

2. Unzip the file you copied to your hard drive (see Prerequisites).

3. Configure access to the new database by editing the unzipped file conf/flyway.conf, targeting either the Core Database or the database you created in Step 1. Specifically, specify the following parameters in that file:

   - flyway.url = JDBC URL format
   - flyway.user = user
   - flyway.password = password
   - flyway.outOfOrder = true
   - flyway.validateOnMigrate = false

4. Do one of the following:

   - If you are using Linux, run the unzipped script ./flyway.sh migrate. This command will target the database you configured in Step 3.
   - If you are using Windows, run the unzipped script flyway.cmd migrate. This command will target the database you configured in Step 3.
5. Verify that the script executed successfully.

   This procedure ends here.
APPENDIX B: Configuring Logging

This appendix describes:

- Enabling Centralized Apache log4j Logging (page 291)
- Logging for PDF Document Engines (page 292)

Enabling Centralized Apache log4j Logging

For detailed information about the Apache log4j logging facility, go to:

- http://logging.apache.org/log4j/1.2/

If OneSpan Sign is deployed on Linux, you can enable centralized logging. That is, you can configure multiple OneSpan Sign nodes running on different JVMs to log to the same file.

Prerequisites

- OneSpan Sign is deployed on Linux in a clustered environment.
- OneSpan Sign has been configured for Apache log4j logging, using the relevant steps from the procedure Installing the Core Component in Chapter 8.

Action

To enable centralized logging, do the following FOR EACH ONESPAN SIGN NODE:

1. If you are using WebSphere, configure Apache log4j logging to log to SYSLOG by doing the following:
   a. Open the file "log4j.properties" in the directory "[eSL_HOME]/logging".
   b. Optional: Edit the following line by replacing the string "localhost" with the server name or IP address of your SYSLOG service:

   \[log4j.appender.SYSLOG.SyslogHost=localhost\]

   c. Edit the following line by replacing the string "Local3" with the relevant SYSLOG Facility:

   \[log4j.appender.SYSLOG.Facility=Local3\]

   Make the value of "Facility" the SYSLOG Facility you want to log to.

   d. Save and close "log4j.properties".
   e. Restart the Application Server.
2. Configure SYSLOG to log to the desired common file (or any other location supported by SYSLOG).

3. If you want to include the clustered instances in the centralized log file, do the following in the SYSLOG configuration file:
   a. Create a new template that includes the \texttt{HOSTNAME} variable. For example:
   \begin{verbatim}
   $template NameOfTemplate,"%timegenerated% SilanisLog
   %HOSTNAME%: msg\n"
   \end{verbatim}

   b. Apply that template to the SYSLOG facility. For example:
   \begin{verbatim}
   local3.* /var/log/esep.log;NameOfTemplate
   \end{verbatim}

   JVM reads logging properties files once, on start-up. If you change any such file after its server has started, you must restart the server before your changes will take effect.

   This procedure ends here.

**Logging for PDF Document Engines**

A step in Configuring a PDF Document Engine (page 131) instructs you to configure logging for the PDF Document Engines. Information about those engines is logged automatically in the following files:

- \texttt{decore.log} (page 292)
- \texttt{multitenancy.log} (page 295)
- \texttt{deaxis.log} (page 296)
- \texttt{deservice.log} (page 298)

**decore.log**

The \texttt{decore.log} file records information about low-level operations performed by the core of the PDF Document Engine. This section describes the following aspects of that file:

- Tenant Environment (page 292)
- Logging Levels (page 293)
- File Location (page 293)
- Maximum File Size (page 294)
- FIPS Mode (page 294)

**Tenant Environment**

As soon as a transaction occurs in a tenant environment, logging for the relevant tenant switches from a file named \texttt{decore.log} to a file named \texttt{decore-<tenant_name>.log}. 
Logging Levels

Several alternative levels of detail can be recorded by the `decore.log` file. These are described in the Values column of Table 11-2 for the variable CoreTrace. You can configure the logging level by changing that parameter (e.g., "CoreTrace=DEBUG" specifies the logging level as DEBUG).

File Location

This section describes the location of the `decore.log` file in the following environments:

- **PDF Document Engine on Windows** (page 293)
- **PDF Document Engine on Linux** (page 293)

**PDF Document Engine on Windows**

If the PDF Document Engine is installed on Windows, the full path to `decore.log` is:

<DE deployment directory>/services/DEService/logs/decore.log

If you want to change the location of this file, you must change the PDF Document Engine’s deployment directory.

**PDF Document Engine on Linux**

Ensure that the Apache user has WRITE access to the directory where `decore.log` resides.

If the PDF Document Engine is installed on Linux, by default the full path to `decore.log` is in the following location:

- $(SILANIS_HOME)/logs/decore.log

If you want to change the location of this file, do the following:

1. If you are using RHEL/CentOS 7:
   a. Find the variable SILANIS_HOME in the file DEMultitenancy.env. That file resides in the directory <DE_deployment directory>/utils.
   b. Set that variable to the desired new location, as follows:
      
      SILANIS_HOME=<desired log path>

The default value of that variable is <DE deployment directory>.
2. If you are using RHEL/CentOS 6:
   a. Find the variable $SILANIS\_HOME$ in the file $\text{silanis.sh}$. That file resides in the directory $<\text{DE deployment directory}>/\text{utils}$.
   b. Set that variable to the desired new location, as follows:
      
      \[
      \text{export } \text{SILANIS\_HOME}=<\text{desired log path}>
      \]

      \text{This procedure ends here.}

**Maximum File Size**

The PDF Document Engine always logs to a file named $\text{decore.log}$.

Normally, a new $\text{decore.log}$ file is created once the size of the file exceeds the value of the parameter $\text{DecoreSize}$ in Table 11-2 (default = 1 MB). However, it may happen that when that size is exceeded, the log buffer still contains data. If so, the new file will be created only after the buffer is completely written to the existing file.

As the new file is created, the existing $\text{decore.log}$ file is renamed. Multiple renaming events produce a sequence of old log files (named $\text{decore.<rotation\_time>.log}$). By default, the maximum permissible number of such files is 20. You can specify a different value by using the parameter $\text{LogMaxCount}$ in Table 11-2. When the maximum number of files is reached, the oldest file is deleted to make room for a new one.

**FIPS Mode**

Immediately after a PDF Document Engine is restarted, before it processes its first request, the engine records the status of "FIPS mode" in $\text{decore.log}$ [see Appendix F].

If the OpenSSL libraries cannot initialize in FIPS mode, or if you have disabled FIPS mode, the recorded status will be $\text{Currently running with FIPS mode: 'OFF'}$. Otherwise, the recorded status will be $\text{Currently running with FIPS mode: 'ON'}$. 
multitenancy.log

The multitenancy.log file is a log file that records information about remote PDF Document Engine properties in a “tenant environment” [see Tenant Environments, Use the Remote Properties Repository (page 309)].

This section describes the following aspects of that file:

- File Location (page 295)
- Logging Levels (page 295)

**File Location**

The location of that file is the same as that of the decore.log file. For details, see decore.log (page 292).

**Logging Levels**

If you are using Linux, you can set the logging level for this file permanently. If you are using Windows, you can set the logging level for this file temporarily until the DEMultitenancy service is stopped.

The valid logging levels are those listed for the CoreTrace parameter in the Values column of Table 11-2.

To set the logging level for the file multitenancy.log:

1. If you are using RHEL/CentOS 6, open the file silanis.sh in the directory <DE_deployment directory>/utils, and edit the line LogLevel=INFO to specify the desired level (if it’s different from INFO).

2. If you are using RHEL/CentOS 7, open the file DEMultitenancy.env in the directory <DE_deployment directory>/utils, and edit the line LogLevel=INFO to specify the desired level (if it’s different from INFO).

3. If you are using Windows:
   a. Open the Service Manager’s Management console.
   c. Right-click that service, and select Properties.
   d. If the service is running, click Stop to stop the service.
      
      ![Reminder]
      You will be warned that the preceding step will also stop the HTTP server (Apache or IIS).

   e. In the Start parameters text box, type the desired logging level.
   f. Click Start to start the service.
g. Start Apache.

The logging level you specified will remain in use until the service is stopped.

This procedure ends here.

**deaxis.log**

The deaxis.log file is a log file that records information about the PDF Document Engine’s AXIS2/C service layer.

The log mechanism is based on the AXIS2/C framework. For more details, consult the documentation at the following Web site:


The remainder of this section describes the following aspects of deaxis.log:

- **File Location (page 296)**
- **PDF Document Engine on Apache (page 296)**
- **PDF Document Engine on IIS (page 297)**

### File Location

If you are using Linux, by default the deaxis.log file is in the following location:

- `<DE deployment directory>/logs/deaxis.log`

If you are using Windows, by default the deaxis.log file is in the following location:

- `<DE deployment directory>\services\DEService\logs\deaxis.log`

You can modify the specified location by editing `httpd.conf` (if you are using Apache) or the Windows Registry (if you are using IIS).

Ensure that the Apache user has WRITE access to the log file’s directory.

### PDF Document Engine on Apache

If the PDF Document Engine is installed on Apache, the location, maximum size, and logging level of the deaxis.log file are specified by the following lines in Apache’s configuration file (generally `httpd.conf`):

```
Axis2LogFile <DE deployment directory>/logs/deaxis.log
Axis2MaxLogFileSize <maximum size of log file>
```
Axis2LogLevel LOG_LEVEL

When the deaxis.log file reaches the specified <maximum size of log file>, it is saved with a new filename (deaxis.<rotation_time>.log), and a new log file is created. Once the new file reaches the maximum size, it replaces the old file. Thus at any given moment, the system maintains at most two log files. Their default maximum size is 1 MB.

The above LOG_LEVEL is a placeholder for one of the following values:

- crit – Logs critical errors only
- error (default) – Logs all errors
- warn – Logs warnings and above
- info – Logs information and above
- debug – Logs debugging information and above
- trace – Logs trace messages

PDF Document Engine on IIS

If the PDF Document Engine is installed on IIS, use the following procedure to configure the location and logging level of the deaxis.log file.

To configure the "deaxis.log" file for a PDF Document Engine on IIS:

1. Ensure that the following key has been added to the Registry:
   
   HKEY_LOCAL_MACHINE/SOFTWARE/Apache Axis2c/IIS ISAPI Redirector

2. Ensure that under this key:
   a. The "axis2c_home" sub-key has been added, and its value is <DE
deployment directory>. This parameter specifies the parent
directory in which the Axis2c lib folder resides. That folder contains the Axis2c third-party libraries.
   b. The file-location sub-key "log_file" has been added, and its
      value is the full path of the deaxis.log file:
      
      log_file = <DE deployment directory>\services\DEServices\logs\deaxis.log
   c. The logging-level sub-key "log_level" has been added, and its
      value is one of the following:
      
      • critical – Logs critical errors only
      • error – Logs all errors
      • warning – Logs warnings and above
      • info – Logs information and above
      • debug (default) – Logs debugging information and above
      • trace – Logs trace messages

      This procedure ends here.

If the PDF Document Engine is installed on IIS, the deaxis.log file’s default
maximum size is 32 MB. When that file reaches its maximum size, it is saved with a new filename (deaxis.<rotation time>.log), and a new log file is created. Once the new file reaches the maximum size, it replaces the old file. Thus at any given moment, the system maintains at most two log files.

**deservice.log**

The deservice.log file is a log file that records information about the PDF Document Engine’s SOAP communications.

This section describes the following aspects of that file:

- Tenant Environment (page 298)
- Logging Levels, File Location, and Maximum File Size (page 298)

**Tenant Environment**

As soon as a transaction occurs in a tenant environment, logging for the relevant tenant switches from a file named deservice.log to a file named deservice-<tenant_name>.log.

**Logging Levels, File Location, and Maximum File Size**

For logging levels, file location, and maximum file size, the file deservice.log uses the same rules and configuration settings as the file decore.log. The only exception is the Table 11-2 parameter LogFilter, which is not relevant for deservice.log.

For details, see decore.log (page 292).
APPENDIX C: Configuring Cryptographic Libraries

This appendix is relevant for all JCE configurations (Table 2-2’s Application Servers, the e-Vault Manager, etc.).

To configure cryptographic libraries for an Application Server:

1. Install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files for your version of the JDK. If you are using WebSphere, these files can be downloaded from IBM’s website. If you are using JBoss, Wildfly, or WebLogic, these files can be downloaded from Oracle’s website.

   To perform the installation, replace the policy files in the JRE’s “\lib\security” folder with the downloaded files.

2. Restart the Application Server.

   This procedure ends here.
APPENDIX D: Expected Log-File Contents

This appendix describes expected content that may appear in OneSpan Sign log files. Specifically, it describes the following types of messages:

- **Real Errors** (page 300)
- **Harmless Messages** (page 301)

### Real Errors

This section describes the following types of real errors:

- **Document Engine Read Timed Out** (page 300)
- **Unable to Distribute Document** (page 301)

### Document Engine Read Timed Out

At any time, an error message could appear that begins with something like the following lines:

```
com.silanis.awsng.document.DocMgr executeAction

WARNING: document engine action failed: try 1/3, pid = -1, caller = customizeDocument, action = [action = CustomizeAction, name = null, doc. name = 1_M316NJ(03-10), new doc. name = 1_M316NJ(03-10)_customized], command = [id = 1_M316NJ(03-10)_customized

processId = -1
userId = 0
documentId = 0
documentName = null
description = null
pageWidth = 0
pageHeight = 0
}

pid = -1

extra = com.silanis.docengine.command.CommandException:
com.silanis.docengine.command.router.RouterException:
java.net.SocketTimeoutException: Read timed out
```

This exception is not harmless. If it appears, increase the PDF Document Engine parameter `aws.document.engine.timeout (milliseconds)`; see the Platform Administrator’s Guide’s Table C-20 (Settings for Doc Engines in "documentEngine.properties"). If, for example, `aws.document.engine.timeout (milliseconds)` has its default value of 30000, increase it to at least 60000 (1 minute).
Unable to Distribute Document

After the system tries to execute an `ActionDistribute`, `action-distribute`, or `template-action-distribute` command, an "Unable to distribute document" error message will appear if either or both of the following is true:

- The destination directory is on a different Operating System from the one where OneSpan Sign is installed.
- The destination directory’s permissions don’t permit renaming or deleting files.

For example, the error message could contain something like the following lines:

```java
Caused by: com.silanis.awsng.process.ProcessRuntimeException: Unable to distribute document
... 
Caused by: java.io.IOException: The file access permissions do not allow the specified action.
at java.io.File.checkAndCreate(File.java:1716)
at java.io.File.createTempFile0(File.java:1737)
at java.io.File.createTempFile(File.java:1813)
at com.silanis.awsng.action.distribute.DistributeAction.
directoryDistribute(DistributeAction.java:265)
```

If such a message appears, add the following JVM argument to the Application Server that runs OneSpan Sign:

```
distribute_without_temp_file = true
```

Harmless Messages

You may find that the OneSpan Sign home directory contains a harmless file called "sent_warnings.properties". That file is used to send the Sys Admin warnings about the PDF Document Engine (e.g., A certificate has expired).

This section describes the following types of harmless messages:

- Startup with an Empty Core Database (page 302)
- Message After Restarting the Application Server (page 302)
- Redundant Core Group Policies (page 302)
- Class Path Contains Multiple SLF4J Bindings (page 302)
- Class Path Contains Two JAXB Versions (page 303)
- Could Not Synchronize Database State with Session (page 303)
- Batch Update Exception (page 304)
- Message Caused by an HTTP Call (page 304)
- Signer Certificate Not in Properties File (page 305)
**Startup with an Empty Core Database**

If you start or restart the Application Server when the Core Database is empty, a harmless error message may appear that begins with the following lines:

```
Error getting root configuration: "Cannot find default configuration, root node not defined.".
```

```
com.silanis.idmgr.request.exception.RequestFailureException: Cannot find default configuration, root node not defined.
```

**Message After Restarting the Application Server**

Immediately after restarting the Application Server, a harmless Xalan-related exception may appear that begins with the following lines:

```
java.lang.NoSuchMethodException:
org.apache.xpath.compiler.FunctionTable.installFunction
```
```
java.lang.NoSuchFieldException: m_functions
```
```
at java.lang.Class.getField(Class.java:1520)
```
```
at org.apache.xml.security.Init.registerHereFunction(Unknown Source)
```
```
at org.apache.xml.security.Init.init(Unknown Source)
```

**Redundant Core Group Policies**

After starting a cluster that has redundant Core Group Policies (probably after using the Installer to uninstall and re-install), a harmless warning may appear that begins with the something like the following lines:

```
HAManagerImpl WHMGR0302W: Multiple policies match the group named IBM_hc=awsCluster,WSAF_SIB_BUS=esepBus,WSAF_SIB_MESSAGING_ENGINE= XXX
```

To remove that message:

1. In the WebSphere Admin Console, click Buses > esepBus > Bus members > awsCluster > Redundant core group policies. You will see two policies that match the above description.
2. Remove the redundant policy.
3. Restart the server.

This procedure ends here.

**Class Path Contains Multiple SLF4J Bindings**

A harmless message related to SLF4J bindings may appear if any of the following occurs: (1) the Application Server is restarted; (2) the Core Component is installed; (3) the e-Mail Notification ManagerTM is installed.

That message includes something like the following lines:

```
Class path contains multiple SLF4J bindings.
```
Class Path Contains Two JAXB Versions

If you are using WebSphere, and if the Application Server is restarted, a harmless message may appear that is related to the use by OneSpan Sign of two JAXB versions.

That message begins with something like the following lines:

WSSERVLET11: failed to parse runtime descriptor: java.lang.LinkageError: You are trying to run JAXB 2.0

runtime (from wsjar:file:/usr/IBM/WebSphere8/profiles/AppSrv01/installedApps/bellerophonNode01Cell/eSL-Core.ear/library/axis2/jaxb-impl-2.1.7.jar!/com/sun/xml/bind/v2/model/impl/ModelBuilder.class) but you have old JAXB 1.0 runtime earlier in the classpath (at wsjar:file:/usr/IBM/WebSphere8/profiles/AppSrv01/installedApps/bellerophonNode01Cell/eSL-Core.ear/library/jaxb-impl.jar!/com/sun/xml/bind/WhiteSpaceProcessor.class) Please remove the JAXB 1.0 runtime for 2.0 runtime to work correctly.

java.lang.LinkageError: You are trying to run JAXB 2.0 runtime (from wsjar:file:/usr/IBM/WebSphere8/profiles/AppSrv01/installedApps/bellerophonNode01Cell/eSL-Core.ear/library/axis2/jaxb-impl-2.1.7.jar!/com/sun/xml/bind/v2/model/impl/ModelBuilder.class) but you have old JAXB 1.0 runtime earlier in the classpath (at wsjar:file:/usr/IBM/WebSphere8/profiles/AppSrv01/installedApps/bellerophonNode01Cell/eSL-Core.ear/library/jaxb-impl.jar!/com/sun/xml/bind/WhiteSpaceProcessor.class) Please remove the JAXB 1.0 runtime for 2.0 runtime to work correctly.

Could Not Synchronize Database State with Session

At any time, a harmless message could appear that begins with something like the following lines:


org.hibernate.StaleObjectStateException: Row was updated or deleted by another transaction (or unsaved-value mapping was incorrect): [com.silanis.awsng.hibernate.AwsProcessSession#10296308]

This exception appears because OneSpan Sign sometimes needs more than one attempt to synchronize with its database. If the number of attempts is fewer than 5,
Batch Update Exception

At any time, a harmless message could appear that begins with something like the following lines:

2011-04-13 12:55:56,843 ERROR

org.hibernate.exception.ConstraintViolationException: Could not execute JDBC batch update
at org.hibernate.exception.SQLStateConverter.
    convert(SQLStateConverter.java:71)
Caused by: java.sql.BatchUpdateException: ORA-00001: unique constraint (AWS384USER.SYS_C0044112) violated

This exception appears because OneSpan Sign sometimes needs more than one attempt to synchronize with its database in a batch operation. If the number of attempts is fewer than 5, such a message is expected.

Message Caused by an HTTP Call

When the PDF Document Engine cannot handle an HTTP call that was not intended for it, a harmless message like the following may appear in the "deservice.log" file:

[Thu Jan 10 07:20:23 2013] [error].\.\..\src\core\transport\http\util\http_transport_utils.c(2557) Service or operation not found

HTTP calls like the following will generate such a message:

- http(s)://localhost/axis2/services
- http(s)://localhost/axis2/
- http(s)://localhost – This simple call generates a harmless message when the PDF Document Engine uses IIS, but not when it uses Apache.
**Signer Certificate Not in Properties File**

If a tenant other than the System Tenant signs a document, the following harmless message is written into the file `decore.log`:

```
ERROR: Signer certificate is not in properties file.
```

**Note:** The tenant could not have signed if the Signer Certificate was missing.
APPENDIX E: Creating a Keystore

This appendix is referenced in Chapter 12 and Chapter 18. It describes how to create a keystore for your production environment using one or more certificates from a Certificate Authority (CA) such as Verisign.

Prerequisites

- You have created a private/public key pair.
- Your private key is in PEM format.
- Using the key pair, you have created a Certificate Signing Request, and have sent it to a CA.
- In response, you have received leaf, intermediate, and root certificates from the CA.
- The Open Source tool called openssl is installed on your system.

Action

To create a PKCS12 or JKS keystore with certificates from a Certificate Authority:

1. Open a command shell.
2. If one or more CA-issued certificates is in CER or CRT format, convert each file to PEM format by running the following command for each file:

   openssl x509 -in [original cert] -inform DER -out [cert.pem]

   Replace [original cert] with the name of the input CER or CRT file, and replace [cert.pem] with the name you want for the output PEM file.

3. Combine the CA-issued certificates into a single file by running the following command:

   cat [leaf.pem] [intermediate_ca.pem] [root_ca.pem] >> clientcertchain.pem

   Here: (1) [leaf.pem] is the PEM file that contains the leaf certificate received from the CA; (2) [intermediate_ca.pem] is the PEM file that contains the intermediate certificates received from the CA; (3) [root_ca.pem] is the PEM file that contains the root certificate received from the CA.

4. Create a PKCS12 keystore as follows:

   a. Run the following command:

      openssl pkcs12 -export -in clientcertchain.pem -inkey [private-key].pem -out [keystore].pfx -alias [key alias]

      Here: (1) clientcertchain.pem is the file created in the previous step; (2) [private-key].pem is the name of your private-key file; (3) [keystore].pfx is the name you want for the output keystore; (4) [key alias] is the alias you want for the private-key entry in the keystore.
b. When prompted, enter the password of your private-key file.
c. When prompted, enter a password of your choice for the newly created keystore.

5. If you want a JKS keystore, use the *Java Keytool* to convert into a JKS keystore the PKCS12 keystore you just created.

   This procedure ends here.
APPENDIX F: FIPS 140-2 Compliance

This appendix is relevant only if your organization must comply with FIPS 140-2.

FIPS is an abbreviation for Federal Information Processing Standards. Those standards are issued by the U.S. National Institute of Standards and Technology (NIST). For more, see:


The PDF Document Engine runs in "FIPS mode" by default, but it can be configured not to do so.

That mode entails that:

- Every digital signature created by OneSpan Sign will comply with FIPS 140-2.
- If you sign with a certificate, it must be FIPS-compliant. That is, it must use compliant algorithms with sufficient key lengths, and it must not be self-signed.
- The PDF Document Engine uses the cryptographic library OpenSSL FIPS Object Module, which is FIPS 140-2 certified.

If your organization must comply with FIPS 140-2, you should ensure that each instance of the PDF Document Engine is installed in FIPS mode (true by default).

To enable OneSpan Sign to create signatures that comply with FIPS 140-2, OneSpan Sign uses algorithms that comply with FIPS 140-2. For example, all signatures are created using SHA-256.

OneSpan Sign encrypts sensitive data in its databases (see, for example, the encryption.properties Resource Key in Table 18-1). Such encryption can optionally be done with FIPS-compliant algorithms. The procedure required to apply those algorithms depends on your Application Server type (WebSphere, JBoss, Wildfly, WebLogic). To view the relevant procedure for your server, consult the server’s vendor documentation or contact OneSpan.
APPENDIX G: Tenant Environments

The section Tenant and Non-Tenant Environments (page 112) introduces the concept of a "tenant environment" for the PDF Document Engine.

The following sections provide more information about tenant environments:

- Tenant Environments Use the Remote Properties Repository (page 309)
- Propagating Properties from the Remote Properties Repository (page 310)
- Logging in a Tenant Environment (page 311)

Tenant Environments Use the Remote Properties Repository

If the PDF Document Engine is running in a tenant environment, it fetches its properties from the Remote Properties Repository.

How this feature is configured depends on whether your PDF Document Engine’s operating system is Linux or Windows.

If the PDF Document Engine is running on Linux:

- This feature is implemented via an executable file called DEMultitenancy.
- This file is installed in the directory <DE installation folder>/silanis.
- This file is a standard application.

If the PDF Document Engine is running on Windows:

- This feature is implemented via an executable file called DEMultitenancy.exe.
- This file is installed in the directory <DE installation folder>/services\DEService.
- This file is a Windows Service Application.
- When you are Installing a New Doc Engine on Windows with Apache (page 123), if you are installing in a tenant environment, you must select the option Remote Document Engine Properties. If you do so, the Installer will register the Windows Service Silanis e-SE Remote Document Engine Properties, and it will create a dependency between that service and the hosting service (Apache or IIS). The following screen shows exactly that dependency for Apache 2.2.
If the PDF Document Engine is running in a "tenant environment", and you want to change some of its properties, you must do so via the Admin Console [see Tenant and Non-Tenant Environments (page 112)]. This will change those properties in the Remote Properties Repository.

To propagate those changes from the Remote Properties Repository to your PDF Document Engines, you must perform the following procedure.

**To propagate changed properties from the Remote Properties Repository to individual PDF Document Engines:**

1. If the PDF Document Engine is running on Linux, restart Apache.
2. If the PDF Document Engine is running on Windows:
   a. Stop the **Silanis e-SE Remote Document Engine Properties** service.
   b. Start the **Windows Service Manager** (execute services.msc).
   c. Select **Silanis e-SE Remote Document Engine Properties**, and click the **Stop** button.

   An alternative way of stopping the service is to run the following command on a command line: `net stop "Silanis e-SE Remote Document Engine Properties"`
d. Start the HTTP server service as follows:

A. From the Windows Service Manager, select the HTTP server service (Apachex.x for Apache; World Wide Web Publishing Service for IIS).

B. Click Start.

C. Do one of the following:
   • If you are using Apache, run the following command:
     net start Apachex.x
   • If you are using IIS, run the following command:
     net start "World Wide Web Publishing Service"

This procedure ends here.

Logging in a Tenant Environment

The executables DEMultitenancy and DEMultitenancy.exe create logging information in a file called multitenancy.log. For more about that file, see FIPS Mode (page 294) in Appendix B.
APPENDIX H: App Server Procedures

A number of chapters reference this appendix, which contains Application Server procedures for doing the following:

- Preparing an App Server Environment (page 312)
- Creating a JDBC Provider (page 316)

The procedures in this appendix assume that your environment meets all relevant requirements in Chapter 2.

Preparing an App Server Environment

If you are preparing a clustered environment, we strongly recommend that you create that environment in accordance with the topologies described in Chapter 2.

If you are preparing a clustered environment, you should use a suitable time-sync service to synchronize the clocks on all cluster members to within one second of each other. For more on this, see: http://www.boulder.nist.gov/timefreq/service/its.htm

The following procedures describe how to create an Application Server environment for all supported Application Server types (see Table 2-2):

- Preparing an Environment for WebSphere (page 312)
- Preparing an Environment for JBoss 6 or Wildfly (page 314)
- Preparing an Environment for WebLogic (page 316)

Preparing an Environment for WebSphere

To prepare an Application Server environment on WebSphere, ensure that the following conditions have been met:

1. A WebSphere profile has been created, and WebSphere administrative security has been enabled.

   The recommended practice is to install OneSpan Sign and all its components on their own WebSphere profile.

2. If there is a previous OneSpan Sign installation, and if you want to change the queue type from WebSphere MQ to System Integration Bus MQ resources (or vice versa), then all existing queues, activation specifications, and queue factories related to OneSpan Sign have been manually removed.

3. If you are using WebSphere MQ messaging and you plan to install the Core Component, the following persistent queues have been created on the MQ server:
   - APPROVEIT.ERROR.QUEUE
4. If you are using WebSphere MQ messaging and you plan to install the e-Mail Notification Manager™, the following persistent queues have been created on the MQ server:
   • APPROVEIT.SCHEDULER.QUEUE
   • APPROVEIT.NOTIFIER.REQUEST.QUEUE

5. If you are using WebSphere MQ messaging and you plan to install e-Witness, the following persistent queue has been created on the MQ server:
   • APPROVEIT.PRINTTOPDF.QUEUE

6. If you are using WebSphere MQ messaging and you plan to install the Event Manager, the following persistent queue has been created on the MQ server:
   • APPROVEIT.EVENTS.QUEUE

7. If you are using WebSphere MQ messaging and you plan to install the OneSpan Sign Application, the following persistent queues have been created on the MQ server:
   - statisticsQueue
   - expirePackageQueue
   - packageCallbackQueue
   - connectorsFrameworkCallbackQueue
   - salesforcePackageEventQueue
   - salesforceUploadCompletedQueue
   - evidenceSummaryQueue
   - packageRemindersQueue
   - emailQueue
   - smsQueue
   - asyncBusinessQueue

8. You have configured the logging behavior to use the Apache log4j logging facility by doing the following:
   a. Copy the file "log4j.jar" from the "Server/Core Components/Thirdparty" directory of the software provided by OneSpan into a shared directory to which WebSphere has READ access.
   b. If you are using a Linux Operating System, change the permission on "log4j.jar" so WebSphere Applications will have READ access to it.
   c. Create a shared library at the Cell level for the cell to which your nodes belong.
   d. Add to this shared library’s classpath the full path (including filename) of the newly copied "log4j.jar" file from Step a.
e. Copy the file "aspectjrt-1.8.5.jar" from the "Server/Core Components/Thirdparty" directory of the software provided by OneSpan into a shared directory to which WebSphere has READ access.

f. If you are using a Linux Operating System, change the permission on "aspectjrt-1.8.5.jar" so WebSphere Applications will have READ access to it.

g. Create a shared library at the Cell level for the cell to which your nodes belong.

h. Add to this shared library's classpath the full path (including filename) of the newly copied "aspectjrt-1.8.5.jar" file from Step e.

i. Perform the following related steps:
   • Create a new class loader for your OneSpan Sign server's JVM.
   • Configure this class loader so classes are loaded with parent class loader last.
   • To this class loader, add both shared libraries created in Step c and Step g.

9. If you are using WebSphere 9, configure the Application Server with the following generic Java Virtual Machine (JVM) argument:
   • -Dcom.ibm.ws.cdi.enableCDI=false -agentlib:getClasses

   The above argument must be specified for the Platform and the Application.

10. Before you start deploying Platform or Application software, ensure that the Deployment Manager and all Node Agents are up and running.

    This procedure ends here.

---

**Preparing an Environment for JBoss 6 or Wildfly**

This section discusses:

- Startup Modes (page 314)
- Server Profiles (page 315)
- Prerequisites (page 315)

**Startup Modes**

JBoss 6 and Wildfly have two alternative "startup modes":

- **Standalone Mode** — This is the traditional running mode for a JBoss or Wildfly Application Server. Unlike Domain Mode, it doesn't have any runtime domain-management features. The full path to the startup script for Standalone Mode is `bin/standalone.sh`. Run that script to
activate this mode.

- **Domain Mode** — This mode was introduced in JBoss EAP 6 and Wildfly 7. A domain is a group of servers that can share deployment and configuration information.

⚠️ **WE DO NOT SUPPORT** DOMAIN MODE.

### Server Profiles

A server profile is a set of sub-systems with various functionalities that can be added to an Application Server.

Both JBoss 6 and Wildfly startup modes support a profile called the Full Profile. Indeed, both the OneSpan Sign Platform and OneSpan Sign Application require this profile.

### Prerequisites

**To prepare an Application Server environment on JBoss 6 or Wildfly, ensure that the following conditions have been met:**

1. Before you perform the procedures in this guide, you have chosen a startup mode. See Startup Modes (page 314).
2. You have ensured that all procedures in this guide are applied to a Full Profile configuration file. The default version of such a file is called "standalone-full.xml". Chapter 3’s Installer uses a Full Profile configuration file called "esep.xml" that was created by customizing "standalone-full.xml". To start a JBoss 6 or Wildfly Application Server with the Installer’s "esep.xml" file (or with a similar file of the same name), run the startup script with the following parameter:
   ```
   --server-config=esep.xml
   ```
3. To ensure that the log files are properly created and contain ALL messages generated by OneSpan Sign, you have ensured that the product will be installed using a standalone-full profile or a domain-full profile (two possible Full Profiles).
4. You have ensured that HTTPS transport is enabled. The Administration Console and e-Witness won’t work without it.
5. You have ensured that the Application Server is shut down before you start deploying Platform or Application software.
6. If you are using Wildfly, configure the WELD Sub-System by adding the following properties:
   - `require-bean-descriptor="true"
   - `non-portable-mode="true"

This procedure ends here.
Preparing an Environment for WebLogic

To prepare an Application Server environment on WebLogic, ensure that the following conditions have been met:

1. A WebLogic domain has been created, with at least one WebLogic Server instance assigned the role of the Administrative Server. The domain is WebLogic’s central configuration entity. In creating the domain, follow the recommended topology in Figure 2-4. There should be at least two Managed Server instances to host OneSpan Sign.

   The recommended practice is to install OneSpan Sign and all its components on their own WebLogic domain.

   We recommend that for each domain, you edit WebLogic’s environment script to appropriately configure memory variables for WebLogic’s domain server.

2. To prepare your WebLogic servers for JMS resources, ensure that you: (1) create each JMS Server with an appropriate persistent store; (2) then create an associated JMS Module; (3) then create a Sub Deployment for any previous JMS Module; (4) make all these entities target your Managed Servers appropriately.

   This procedure ends here.

Creating a JDBC Provider

The following procedures describe how to prepare for the creation of a Data Source by creating a JDBC Provider for all supported Application Server types (see Table 2-2):

- Creating a JDBC Provider for WebSphere (page 316)
- Creating a JDBC Provider for JBoss 6 or Wildfly (page 317)
- Creating a JDBC Provider for WebLogic (page 318)

Creating a JDBC Provider for WebSphere

The OneSpan Sign Application’s JDBC provider must be transactional (XA). However, XA transactions are not enabled by default within Microsoft SQL Server. If you are using that server, you must perform a procedure described here: https://msdn.microsoft.com/en-us/library/aa342335.aspx.

To prepare for the creation of a Data Source by creating a JDBC Provider:

1. Copy the JDBC Provider driver by locating the directory "Server/Core Components/Thirdparty/jdbc_drivers/" provided by OneSpan, and doing one of the following for each Application Server where the Data Source will be installed:
   - If you are using an Oracle database and WebSphere 8, copy the
Creating a JDBC Provider for JBoss 6 or Wildfly

To prepare for the creation of a Data Source by creating a JDBC Provider:

1. Copy the JDBC Provider driver by locating the directory "Server/Core Components/Thirdparty/JBoss6_module" provided by OneSpan, and doing one of the following for each Application Server where the Data Source will be installed:
   - If you are using a DB2 database, copy the sub-directory "ibm" to the directory "[JBoss_Home]/modules".
   - If you are using an SQL Server database, copy the sub-directory "com" to the directory "[JBoss_Home]/modules".
   - If you are using a MySQL database, copy the sub-directory "mysql" to the directory "[JBoss_Home]/modules".
   - If you are using an Oracle database, copy the sub-directory "oracle" to the directory "[JBoss_Home]/modules".

   Do not use any JDBC driver except those in the directories from the previous bullets.

   This procedure ends here.

2. Create a JDBC Provider that uses the driver copied in Step 1.

3. If you are creating a JDBC Provider for the OneSpan Sign Application Backend, under Implementation class name change the JDBC Provider from oracle.jdbc.pool.OracleConnectionPoolDataSource to oracle.jdbc.xa.client.OracleXADatasource.

   This procedure ends here.
Creating a JDBC Provider for WebLogic

The OneSpan Sign Application’s JDBC provider must be transactional (XA). However, XA transactions are not enabled by default within Microsoft SQL Server. If you are using that server, you must perform a procedure described here: https://msdn.microsoft.com/en-us/library/aa342335.aspx.

To prepare for the creation of a Data Source by creating a JDBC Provider:

1. Locate the directory "Server/Core Components/Thirdparty/jdbc_drivers/" provided by OneSpan, and do one of the following for each Application Server where the Data Source will be created:
   - If you are using an Oracle database, copy the JDBC driver "ojdbc6.jar" to a directory in WebLogic’s classpath.
   - If you are using a DB2 database, copy the JDBC driver "db2/db2jcc.jar" and the files "db2/jcc_license_cisuz.jar" and "db2/jcc_license_cu.jar" to a directory in WebLogic’s classpath.
   - If you are using an SQL Server database, copy the JDBC driver "sqlserver/sqljdbc.jar" to a directory in WebLogic’s classpath.

2. Create a JDBC Provider that uses the driver copied in Step 1.

3. If you are creating a JDBC Provider for the OneSpan Sign Application Backend, under Implementation class name change the JDBC Provider from oracle.jdbc.pool.OracleConnectionPoolDataSource to oracle.jdbc.xa.client.OracleXADatasource.

   This procedure ends here.
APPENDIX I: Configuring IIS & Optimizing Performance

This appendix is referenced in the procedure Installing a New Doc Engine on Windows with IIS (page 126).

Prerequisites

- Your environment meets all relevant requirements in Chapter 2.
- Your PDF Document Engine uses:
  - Microsoft Windows Server 2012 or Microsoft Windows Server 2012 R2
  - IIS 8.x (in this case, do all steps below) or Apache (in this case, do only Step 1 below)

Steps 1-2 below optimize a PDF Document Engine’s performance. Steps 3-8 provide a more general configuration of IIS for that engine.

Action

To optimize the performance of a PDF Document Engine, and to configure IIS for that engine:

1. Optimize performance by configuring TCP (Transmission Control Protocol) parameters as follows:
   a. Run regedit.exe.
   b. Right-click the following key:

   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

   c. Select New > DWORD Value.
   d. In the new window that appears, type MaxUserPort and press Return. This creates an MaxUserPort key.
   e. Right-click that MaxUserPort key, and assign it the value 65534.
   f. Select New > DWORD Value.
   g. In the new window that appears, type TCPTimedWaitDelay and press Return. This creates a TCPTimedWaitDelay key.
   h. Right-click that TCPTimedWaitDelay key, and assign it the value 60.
   i. Select New > DWORD Value.
   j. In the new window that appears, type TCPMaxSendFree and press Return. This creates a TCPMaxSendFree key.
k. Right-click that `TCPMaxSendFree` key, and assign it the value 65535.

If you are using Apache, ignore all remaining steps in this procedure.

2. Optimize performance by configuring thread settings as follows:
   a. Run `regedit.exe`.
   b. Right-click the following key:
      
      `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters`
   
   c. Select **New > DWORD Value**.
   d. In the new window that appears, type `MaxPoolThreads` and press **Return**. This creates a `MaxPoolThreads` key.
   e. Right-click that `MaxPoolThreads` key, and assign it the value 1.

   Step 2 guarantees that Windows will set the minimum number of possible threads per process to be as low as Windows permits (typically 4).

3. In the IIS Manager, under **Actions** in the right pane, select **Change .NET Framework Version**, and select the value **v2.0**.

4. Configure **General** settings for your Application Pool as follows:
   a. In the **IIS Manager**, click **Application Pools** in the left pane. In the center pane, click the application pool used by the [IIS Website] (e.g. `DefaultAppPool`). Then in the right pane, click **Advanced Settings**. A new window appears.
   
   b. In the **General** section:
      
      A. Set **.NET Version** = v2.0
      B. Set **Queue Length** = 10000.
      C. Set **Start Value** = **AlwaysRunning**.

5. Configure **Process Model** settings for your Application Pool as follows:
   a. In the **IIS Manager**, click **Application Pools** in the left pane. In the center pane, click the application pool used by the [IIS Website] (e.g. `DefaultAppPool`). Then in the right pane, click **Advanced Settings**. A new window appears.
   
   b. In the **Process Model** section:
      
      A. Set **Maximum Worker Processes** = [(number of cores) + 1]. The number of CPU cores is a property of your Windows machine.
      B. Set **Shutdown Time Limit** = [a large number of seconds]. For example, 9000 seconds = 15 minutes.
      C. Set **Load User Profile** = **True**.
D. Ensure that the **Identity** setting is the IIS user `<domain name><username>`, where `<domain name>` is the DNS name of the IIS server, and `<username>` is the username of the IIS user. The IIS user should have normal user access rights to the server (and have full read and execute access to the PDF Document Engine installation folder).

6. Configure **Rapid Fail Protection** for your Application Pool as follows:
   a. In the **IIS Manager**, click **Application Pools** in the left pane. In the center pane, click the application pool used by the [IIS Website] (e.g. `DefaultAppPool`). Then in the right pane, click **Advanced Settings**. A new window appears.
   b. In the **Rapid-Fail** section, set **Enabled = False**. This is the preferred setting because it avoids the system responding to many "false positives" (e.g., when the port is simply busy).

7. Configure **Recycling** settings for your Application Pool as follows:
   a. In the **IIS Manager**, click **Application Pools** in the left pane. In the center pane, click the application pool used by the [IIS Website] (e.g. DefaultAppPool). Then in the right pane, click **Advanced Settings**. A new window appears.
   b. In the **Recycling** section:
      A. Set **Regular Time Interval** = 1740 (minutes).
      B. Set **Private Memory Limit (KB)** = 1048576.
      C. Set the **Virtual Memory Limit** to the following calculated value:
         
         
         
         `[Total Available RAM (GB) / Maximum Worker Processes] * 1000000`

8. Restart IIS as follows:
   a. In the left pane of the **IIS Manager**, click your home page. Under **Actions** in the right pane, click **Start** or **Restart**
   b. In the left pane of the **IIS Manager**, click **Application Pools**. In the center pane, click the Application Pool you are using (e.g., `DefaultAppPool`). Under **Actions** in the right pane, click **Start** or **Restart**.
   c. In the left pane of the **IIS Manager**, click your chosen Web site. In the right pane, click **Start** or **Restart**.
   d. To verify that IIS has successfully restarted, execute one or two e-signature transactions. If you receive the desired response from the system, you can leave IIS running. Then you can begin load tests, or go into production.

   This procedure ends here.
APPENDIX J: Running the DE as an Application User

This appendix is referenced in the procedures Installing a New PDF Document Engine on Linux (page 114) and Uninstalling PDF Document Engines (page 157).

If the procedure Installing a New PDF Document Engine on Linux (page 114) has been successfully completed in the recommended fashion, the PDF Document Engine can be started on Apache by the superuser. We do not recommend allowing an application user to start it, since Apache will become less secure. If you nonetheless want to do so, continue reading this appendix.

RHEL/CentOS 6 and RHEL/CentOS 7 are the only Linux versions supported by the PDF Document Engine. Depending on which version you’re using, perform the appropriate one of the following procedures:

- Starting the Doc Engine on RHEL/CentOS 6 as an Application User (page 322)
- Starting the Doc Engine on RHEL/CentOS 7 as an Application User (page 323)

This appendix also describes:
- Uninstalling the Doc Engine as an Application User (page 325)

Starting the Doc Engine on RHEL/CentOS 6 as an Application User

Prerequisites

- The procedures Installing a New PDF Document Engine on Linux (page 114) and Configuring a PDF Document Engine (page 131) have been performed.
- The superuser (username = root) has done the following:
  a. From within the "install" directory created in the procedure Installing a New PDF Document Engine on Linux (page 114), execute the following command:

     `/admininstall6.sh <user> <group>

     Here <user> is a placeholder for the relevant application user’s username, and <group> is a placeholder for the name of their group. This application user does not need to be the one who participated in the procedure Installing a New PDF Document Engine on Linux (page 114), but it could be.

You can run this script either as a PDF Document Engine server boot-up procedure (recommended), or run it once interactively later. If you do the latter, the application user must log in again after the script is run.
b. Run `visudo` to add the following lines to the file `/etc/sudoers`
   (do this step only once):

   ```
   <user> ALL = (ALL) NOPASSWD: /sbin/service
   <user> ALL = (ALL) NOPASSWD: /sbin/initctl
   ```

**Action**

To start the PDF Document Engine on RHEL/CentOS 6 as an application user, that user should do the following:

1. Verify that the superuser completed their prerequisite above.
2. In the Apache configuration file (default = `/etc/httpd/conf/httpd.conf`):
   a. Ensure that the "user" and "group" parameters are specified appropriately for the application user who participated in the procedure Installing a New PDF Document Engine on Linux (page 114). This does *not* need to be the application user specified in this procedure’s prerequisites, but it could be.
   b. Set to **8080** the *Listen* port in your firewall to be used by the PDF Document Engine.

   ![Port 8080 is a public port, so it can be accessed by an application user. Port 80, for example, can be used only by the superuser.]

3. In Apache’s SSL configuration file (default = `/etc/httpd/conf.d/ssl.conf`), set to **8443** the port to be used by the OneSpan Sign Application Frontend.

   ![Port 8443 is a public port, so it can be accessed by an application user. Port 443, for example, can be used only by the superuser.]

4. When the application user specified in the prerequisites is ready to do so, they can start Apache as a service (along with DEMultitenancy) by running the command "service httpd start".

   ![The application user can also stop Apache as a service (along with DEMultitenancy) by running the command "service httpd stop".]

This procedure ends here.

**Starting the Doc Engine on RHEL/CentOS 7 as an Application User**

**Prerequisites**

- The procedures Installing a New PDF Document Engine on Linux (page 114) and Configuring a PDF Document Engine (page 131) have
been performed.

- The superuser (username = root) did the following:
  
a. From within the "install" directory created in the procedure Installing a New PDF Document Engine on Linux (page 114), execute the following command:

   ```
   ./admininstall7.sh <user> <group>
   ```

   Here `<user>` is a placeholder for the relevant application user’s username, and `<group>` is a placeholder for the name of their group.

You can run this script either as a PDF Document Engine server boot-up procedure (recommended), or run it once interactively later. If you do the latter, the application user must log in again after the script is run.

b. Run `visudo` to add the following line to the file `/etc/sudoers` (do this step only once):

   `<user> ALL = (ALL) NOPASSWD: /usr/bin/systemctl`

**Action**

To start the PDF Document Engine on RHEL/CentOS 7 as an application user, that user should do the following:

1. Verify that the superuser completed their prerequisite above.
2. Copy the files `<DE deployment directory>/utils/DEHttpd.service` and `<DE deployment directory>/utils/DEMultitenancy.service` to the services directory of the Operating System (e.g., `/usr/lib/systemd/system`).
3. To ensure that all service configurations are up-to-date, run the command "systemctl daemon-reload".
4. In the Apache configuration file (default = `/etc/httpd/conf/httpd.conf`):
   a. Ensure that the "user" and "group" parameters are specified appropriately for the application user who participated in the procedure Installing a New PDF Document Engine on Linux (page 114). This does not need to be the application user specified in this procedure’s prerequisites, but it could be.
   b. Set to **8080** the `Listen` port in your firewall to be used by the PDF Document Engine.

   Port 8080 is a public port, so it can be accessed by an application user. Port 80, for example, can be used only by the superuser.
5. In Apache’s SSL configuration file (default = /etc/httpd/conf.d/ssl.conf), set to 8443 the port to be used by the OneSpan Sign Application Frontend.

Port 8443 is a public port, so it can be accessed by an application user. Port 443, for example, can be used only by the superuser.

6. When the application user specified in the prerequisites is ready to do so, they can start Apache as a service (along with DEMultitenancy) by running the command "systemctl start DEHttpd".

The application user can also stop Apache as a service (along with DEMultitenancy) by running the command "systemctl stop DEHttpd".

This procedure ends here.

Uninstalling the Doc Engine as an Application User

Prerequisites

• The application user performed one of the following procedures:
  • Starting the Doc Engine on RHEL/CentOS 6 as an Application User (page 322)
  • Starting the Doc Engine on RHEL/CentOS 7 as an Application User (page 323)
• The superuser (username = root) ran visudo to add the following line to the file /etc/sudoers (and they did this only once):

  \(<user> \text{ ALL = (ALL) NOPASSWD: /bin/chmod 777 -R repo} \)

This line applies only to the directory repo.

Action

To uninstall the PDF Document Engine as an application user, that user should do the following:

1. From within the install directory, run the following command (replacing the placeholder in angle brackets):

   \(./de_uninstall.sh <DE\ deployment\ directory>\)

2. If you are using RHEL/CentOS 6:
   a. Undo any modifications to the file /etc/init.d/httpd done during installation.
b. Remove the file `silanis.conf` from the `conf.d` sub-folder of your Apache installation (default = `/etc/httpd/conf.d` for a normal `yum` installation).

c. Delete the file `/etc/init/DEMultitenancy.conf`.

3. If you are using RHEL/CentOS 7, delete the files `DEHttpd.service` and `DEMultitenancy.service` from the services directory of the Operating System (e.g., `/usr/lib/systemd/system`).

   This procedure ends here.
# APPENDIX K: Glossary

<table>
<thead>
<tr>
<th>TERM</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADAM</td>
<td>Abbreviation for <em>AWS Database Archival Module</em> (see <em>Archival Module</em>)</td>
</tr>
<tr>
<td>Admin Console</td>
<td>See <em>Administration Console</em>.</td>
</tr>
<tr>
<td>Administration Console</td>
<td>A module that enables <em>OneSpan Sign</em> Administrators to configure and maintain the <em>OneSpan Sign Platform</em></td>
</tr>
<tr>
<td>Adobe Signature</td>
<td>A signature created using the Adobe approach. Such signatures are verifiable using Adobe Acrobat or Adobe Reader, which displays the Adobe Audit Trail.</td>
</tr>
<tr>
<td>Application</td>
<td>See <em>OneSpan Sign Application</em>.</td>
</tr>
<tr>
<td>Archival Module</td>
<td>A component that enables administrators to archive data from and restore data to the Core Database</td>
</tr>
<tr>
<td>Archive Database</td>
<td>A database used by the Archival Module to archive a Production Database</td>
</tr>
<tr>
<td>Audit Service</td>
<td>A service that can create reports about the past activities of <em>OneSpan Sign</em> users and Admin Console users</td>
</tr>
<tr>
<td>BackOffice</td>
<td>See <em>OneSpan Sign BackOffice</em>.</td>
</tr>
<tr>
<td>BackOffice Support</td>
<td>A service module that supports the internal functioning of <em>OneSpan Sign BackOffice</em></td>
</tr>
<tr>
<td>Classic User Experience (Classic UX)</td>
<td>The Application <em>User Experience</em> (UX) for signers, senders and administrators that became available in <em>OneSpan Sign</em> 6.0, and is still available. See also <em>New User Experience</em>.</td>
</tr>
<tr>
<td>Connector Framework</td>
<td>A standalone Application component that enables individual connectors to connect to <em>OneSpan Sign</em></td>
</tr>
<tr>
<td>Core Component</td>
<td>The primary component that supplies the Platform’s core functionality (e.g., coordinate transactions and events, access the Data Model). It runs on an Application Server.</td>
</tr>
<tr>
<td>Core Database</td>
<td>A database used by the Core Component</td>
</tr>
<tr>
<td>Electronic Evidence™</td>
<td>Digital evidence related to an e-signature process, which <em>OneSpan Sign</em> collects and securely stores within the process. This evidence has legal significance, and can be used in a court of law.</td>
</tr>
<tr>
<td>Electronic Evidence™ Export Utility</td>
<td>An e-Witness utility that, for a specified e-signature process, prints into a PDF file or a Web Archive HTML file all the screens that would be displayed by e-Witness</td>
</tr>
<tr>
<td>e-Mail Notification Manager™</td>
<td>A component that sends e-Mail invitations and reminders to sign documents</td>
</tr>
<tr>
<td>e-Mail Notification Manager Database</td>
<td>A database used by the Core Component and the e-Mail Notification Manager™</td>
</tr>
</tbody>
</table>

© 2020 - OneSpan North America Inc.    DG-OSS-72 - All rights reserved. 327
<table>
<thead>
<tr>
<th>TERM</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>eSignLive for Box</td>
<td>An Application connector that enables OneSpan Sign users to obtain securely signed Box documents. This term is scheduled to become OneSpan Sign for Box.</td>
</tr>
<tr>
<td>eSignLive for Microsoft Dynamics CRM</td>
<td>An Application connector that enables OneSpan Sign users to obtain securely signed Microsoft Dynamics CRM documents. This term is scheduled to become OneSpan Sign for Microsoft Dynamics CRM.</td>
</tr>
<tr>
<td>eSignLive Print Driver</td>
<td>An Application connector that enables documents to be printed directly into an Application package from any print-enabled Windows application (e.g., Word, Excel, Adobe Reader). This term is scheduled to become OneSpan Sign Print Driver.</td>
</tr>
<tr>
<td>e-Vault Manager™</td>
<td>A companion server product that can create, control, and transfer ownership of the financial instruments called Transferable Records (e.g., e-Notes in a mortgage process)</td>
</tr>
<tr>
<td>Event Manager</td>
<td>A module that monitors internal events</td>
</tr>
<tr>
<td>e-Witness™</td>
<td>A module that enables authorized parties to find, review, and verify a stored e-signature process. It also provides a utility to securely export Electronic Evidence™ to a PDF file and/or a Web Archive HTML file. Before OneSpan Sign 5.1, e-Witness was known as the Process Reviewer.</td>
</tr>
<tr>
<td>Hardware Security Module (HSM)</td>
<td>A standalone physical module that safeguards and manages digital keys for strong authentication, provides cryptographic encoding/decoding of signer certificates, and isolates private keys from network attacks</td>
</tr>
<tr>
<td>HSM</td>
<td>See Hardware Security Module.</td>
</tr>
<tr>
<td>New User Experience (New UX)</td>
<td>The Application User Experience (UX) for senders and administrators that became available in OneSpan Sign 6.1. For sender and administrator actions, this UX represents an enhancement over the Classic UX.</td>
</tr>
<tr>
<td>Notifier</td>
<td>A component of the e-Mail Notification Manager™ that sends notifications. Important examples of such notifications are e-Mail warnings sent to the OneSpan Sign Administrator about the PDF Document Engine (e.g., A certificate is about to expire).</td>
</tr>
<tr>
<td>OneSpan Sign</td>
<td>Software that provides a complete E-Signature Process Management platform for the Web, including preparing, distributing, reviewing, signing, and downloading documents</td>
</tr>
<tr>
<td>OneSpan Sign Application</td>
<td>This is essentially the front end that before OneSpan Sign 6.0 was available only via a SaaS deployment. Users, administrators, and integrators can interact with the Application using its GUI, its API, its SDKs, or its connectors.</td>
</tr>
<tr>
<td>TERM</td>
<td>DEFINITION</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>OneSpan Sign Application Backend</td>
<td>Back-end support for the REST Integration Model (REST API, Java &amp; .Net SDKs) and the <em>OneSpan Sign Application</em></td>
</tr>
</tbody>
</table>
| OneSpan Sign Application Frontend | The Application’s *User Experience* (UX). In fact, *OneSpan Sign* offers two UXs:  
  - The *Classic User Experience* for signers, senders and administrators. This UX became available in *OneSpan Sign 6.0*, and is still available.  
  - The *New User Experience* for senders and administrators. For sender and administrator actions, this UX represents an enhancement over the *Classic UX*. It became available in *OneSpan Sign 6.1*. |
<p>| OneSpan Sign Authentication | An Application tool that authenticates user accounts |
| OneSpan Sign BackOffice | A module that enables <em>OneSpan Sign</em> Administrators to configure and maintain the <em>OneSpan Sign Application</em> |
| OneSpan Sign Controller | An Application component that enables communication between the Platform and the Application |
| OneSpan Sign Document Converter | An Application module that converts a document from DOC or ODT format to PDF format |
| OneSpan Sign e-Notary Validation | An Application tool that potentially enables the online validation of notaries |
| OneSpan Sign Equifax | An Application tool that authenticates user accounts created by <em>OneSpan Sign BackOffice</em> using Equifax |
| OneSpan Sign for Microsoft SharePoint | An Application connector that enables <em>OneSpan Sign</em> users to obtain securely signed Microsoft SharePoint documents |
| OneSpan Sign for Salesforce | An Application connector that enables <em>OneSpan Sign</em> users to obtain securely signed Salesforce documents |
| OneSpan Sign Home Directory | The main directory used by <em>OneSpan Sign</em> (this directory contains the <em>OneSpan Sign</em> license file, for example) |
| OneSpan Sign Mobile SDK | An Application feature that enables users to run e-signature processes from mobile devices |
| OneSpan Sign Platform | This is the core transactional and business logic of <em>OneSpan Sign</em>. |
| OneSpan Sign Scheduler | An Application component that schedules a notification to be sent at a future time |
| OneSpan Sign Single Sign-On | An Application tool that authenticates user accounts against a supported external Authorization Server/Identity Provider |
| OneSpan Sign URL Mapper | An Application tool that maps shortened URLs to complete Application URLs |</p>
<table>
<thead>
<tr>
<th>TERM</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDF Document Engine</td>
<td>A component that performs all operations that manipulate a PDF document (signing, etc.)</td>
</tr>
<tr>
<td></td>
<td>Note the following distinction:</td>
</tr>
<tr>
<td></td>
<td>• Physical PDF Document Engine — A single installation of a PDF Document Engine on a given machine</td>
</tr>
<tr>
<td></td>
<td>• Logical PDF Document Engine — A process on a physical PDF Document Engine that independently handles PDF-document-related commands from OneSpan Sign</td>
</tr>
<tr>
<td>PEM file</td>
<td>A file with the extension pem (Privacy Enhanced Mail) that contains a private key or one or more certificates used during the signing or tampersealing of PDF documents. The private key and certificates used by the PDF Document Engine are in multiple PEM files.</td>
</tr>
<tr>
<td>Personal Certificate Client</td>
<td>An Application module used to sign documents with a personal digital certificate</td>
</tr>
<tr>
<td>Platform</td>
<td>See OneSpan Sign Platform.</td>
</tr>
<tr>
<td>Proxy Server</td>
<td>An OneSpan Sign module that serves as the system’s gatekeeper for all outgoing traffic to the Internet</td>
</tr>
<tr>
<td>Resource Manager</td>
<td>A module that manages resources (e.g., templates, properties, Data Sources)</td>
</tr>
<tr>
<td>REST Integration Model</td>
<td>As of OneSpan Sign 6.5, this is the only supported integration model. It uses a REST API, a Java SDK, and a .Net SDK.</td>
</tr>
<tr>
<td>Scheduler</td>
<td>A component of the e-Mail Notification Manager™ that schedules a notification to be sent at a future time</td>
</tr>
<tr>
<td>Signature Block</td>
<td>A rectangular space within an electronic document whose purpose is to receive and display an electronic signature</td>
</tr>
<tr>
<td>System Tenant</td>
<td>A tenant called system. A System Admin can perform actions that apply at the System Tenant level, while a Tenant Admin can perform actions that apply to their particular tenant. When a tenant is created, it inherits properties from the System Tenant.</td>
</tr>
<tr>
<td>Tenant</td>
<td>A group of users who share the same view of the software they use. A multitenant environment is one in which a single instance of the software runs on a server that serves multiple tenants. Each tenant is given a dedicated share of the instance (including its own data, configuration, user management, etc.).</td>
</tr>
<tr>
<td>Third-Party Components</td>
<td>Software components not provided by OneSpan (the company), but required for OneSpan Sign (the product)</td>
</tr>
<tr>
<td>Transaction Status API</td>
<td>An API that enables integrators to build an external GUI to monitor and create reports on OneSpan Sign transactions</td>
</tr>
<tr>
<td>TERM</td>
<td>DEFINITION</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Trusted Service Provider</td>
<td>An Application component that enables signers to provide their own signing certificate, and thus to avoid installing a signing certificate within OneSpan Sign's environment</td>
</tr>
<tr>
<td>User Manager</td>
<td>A module that manages user credentials and permissions</td>
</tr>
<tr>
<td>XSLT Integration Model</td>
<td>This is an integration model based on XSLT workflow templates and a SOAP API. Releases of OneSpan Sign before version 6.5 supported this model for on-premises deployments. It is no longer supported.</td>
</tr>
</tbody>
</table>