Landscape Design and Integration

SAP Mobile Platform 2.3 SP04

DOCUMENT ID: DC01916-01-0234-01
LAST REVISED: March 2014

Copyright © 2014 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase trademarks can be viewed at the Sybase trademarks page at *http://www.sybase.com/detail?id=1011207*. Sybase and the marks listed are trademarks of Sybase, Inc. ® indicates registration in the United States of America.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.

Java and all Java-based marks are trademarks or registered trademarks of Oracle and/or its affiliates in the U.S. and other countries.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

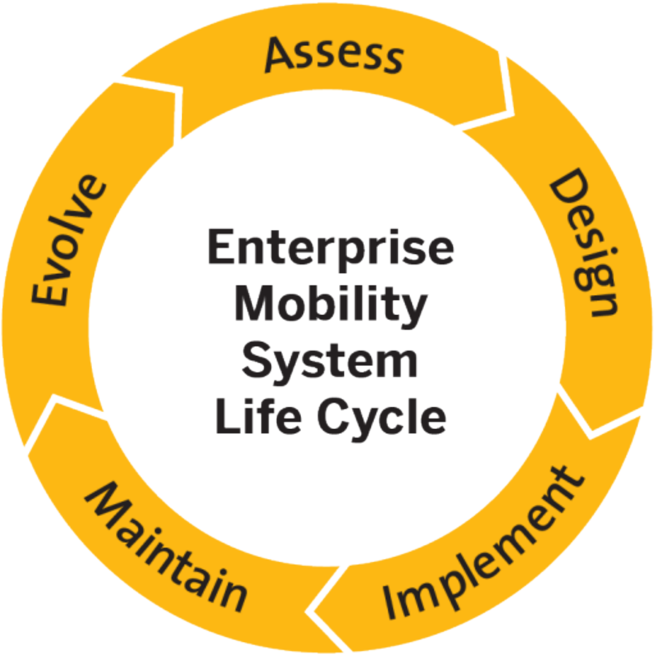
Contents

SAP Mobile Platform System Life Cycle

Use the SAP® Mobile Platform life cycle to define required IT processes for implementing and altering your enterprise mobility management system. Understand each phase so you can successfully deploy and evolve platform components into your ecosystem.

SAP recommends that you review the contents of this document before using the *Installation Guide for Runtime*.

For SAP Mobile Platform, the life cycle is represented by these stages:



Stage 1: Assess

The objective of this stage is to conduct a preliminary analysis of your existing IT ecosystem. Define the mobility goals and required functions and operation of the intended mobile applications and runtime system.

Ecosystem assessments should include:

- Physical plant resources (power, rack space, cable drops, and so on)
- Network design and configuration
- Access control and authentication (information security) mechanisms
- Availability of suitable host systems
- Availability of EIS resources

From the assessment, gather requirements that will help you make design-phase decisions.

Assessing High Availability and Disaster Recovery Objectives

The objectives you set for downtime and data loss largely determine the landscape design for your SAP Mobile Platform implementation.

The main determinants of your SAP Mobile Platform landscape design are the reliability requirements that the system must meet. Two measures of system reliability are typically used:

- Recovery time objective (RTO) – acceptable downtime; the maximum duration of time within which a system must be restored after a failure.
- Recovery point objective (RPO) – acceptable data loss; the maximum tolerable period from which data might be lost due to a major incident.

To help determine how much downtime and data loss is acceptable for your SAP Mobile Platform implementation, consider these items:

- How important are the applications that will be supported?
- What is the cost to your organization when these applications are unavailable?
- What is the cost to your organization of losing data that cannot be restored, and must be reentered?
- Is your organization likely to lose customers if these applications are unavailable, or if customers sometimes have to reenter transactions?

Although it may be difficult to get quantitative answers to such questions, you must determine objectives for downtime and data loss to design your SAP Mobile Platform landscape. Keep these objectives in mind as you review the rest of the assessment topics.

See also

- *Host Platform Requirements* on page 4
- *SAP® Ecosystem Interoperability Requirements* on page 6
- *Network Communications Requirements* on page 7
- *EIS Requirements* on page 10
- *Authentication and Authorization Security Requirements* on page 11
- *Intrusion Detection and Protection Requirements* on page 12

Host Platform Requirements

Provision host systems to meet SAP Mobile Platform Runtime host requirements.

All SAP Mobile Platform Runtime hosts must meet the requirements specified in *Supported Hardware and Software* for:

- Minimum host resources (CPU, RAM, and local storage space)
- Operating system (including edition, version, and service pack)

See also

- *Assessing High Availability and Disaster Recovery Objectives* on page 3
- *SAP® Ecosystem Interoperability Requirements* on page 6
- *Network Communications Requirements* on page 7
- *EIS Requirements* on page 10
- *Authentication and Authorization Security Requirements* on page 11
- *Intrusion Detection and Protection Requirements* on page 12

General Runtime Host Requirements

Guidelines to follow when provisioning hosts for the SAP Mobile Platform Runtime components (SAP Mobile Server and data tier servers) in single-host or clustered landscape designs.

See *Supported Hardware and Software* for basic hardware and operating system requirements. This topic supplements that basic information.

Note: In addition to the resource and OS requirements specified in *Supported Hardware and Software*, the target server cannot include an instance of SAP® Control Center that has been installed for another SAP product.

The following must be identical for all SAP Mobile Platform hosts:

- Product edition and license type
- Processor (32-bit or 64-bit) – you can install the SAP Mobile Platform Enterprise Server Edition only on 64-bit operating systems. You can install the Personal Development Server and Enterprise Development Server Editions on either 32-bit or 64-bit operating systems.
- Operating system – edition, version, and service pack, as well as any intermediate patches or updates
- SAP Mobile Platform software version – including any support package or patch-level updates

See also

- *Data Tier Failover Cluster Host Requirements* on page 5
- *SAP Mobile Server Cluster Host Requirements* on page 5

Data Tier Failover Cluster Host Requirements

Guidelines for requisitioning and setting up the hosts for data tiers in a Microsoft Failover Cluster.

In addition to the general Runtime host requirements:

- All data tier hosts in the cluster must have identical host resources (CPU, RAM, local storage, network and host bus adapters, and so on).
- Host processor architecture must use the same word size (32-bit or 64-bit) as the SAP Mobile Server hosts.
- All data tier hosts must have the appropriate Microsoft Failover Cluster software enabled.
- Follow current Microsoft guidelines for networks, hosts, and storage devices used in a failover cluster.

See also

- *General Runtime Host Requirements* on page 4
- *SAP Mobile Server Cluster Host Requirements* on page 5

SAP Mobile Server Cluster Host Requirements

Guidelines for requisitioning and setting up the hosts for SAP Mobile Server instances in a cluster.

In addition to the general Runtime host requirements:

- Hosts in the SAP Mobile Server cluster need not have identical system resources, but they should have similar processing capabilities, and use the same operating system.
- Host processor architecture must use the same word size (32-bit or 64-bit) as the data tier hosts

See also

- *General Runtime Host Requirements* on page 4
- *Data Tier Failover Cluster Host Requirements* on page 5

SAP® Ecosystem Interoperability Requirements

To share information and services in an SAP landscape, define the degree to which the information and services are to be shared with SAP Mobile Platform. This is a very useful architectural requirement, especially in a complex or extended enterprise.

For SAP Change and Transport System (CTS) requirements, see *Supported LoadRunner Versions*, *CTS Requirements*.

See also

- *Assessing High Availability and Disaster Recovery Objectives* on page 3
- *Host Platform Requirements* on page 4
- *Network Communications Requirements* on page 7
- *EIS Requirements* on page 10
- *Authentication and Authorization Security Requirements* on page 11
- *Intrusion Detection and Protection Requirements* on page 12

SAP SLD Interoperability Requirements

For SAP environments that use Solution Manager for runtime root-cause analysis, configure a destination System Landscape Directory (SLD) server. This configuration allows SAP Mobile Platform to deliver runtime information to a common SAP SLD repository, keeping information about your SAP and SAP Mobile Platform mobility infrastructure complete and current.

To use SAP Mobile Platform data with an SLD:

- **Determine SAP Mobile Server cluster and data tier strategy –** you can configure SLD in a cluster of two: the primary and secondary servers perform different activities, and the cluster database aggregates and holds data. Depending on your schedule and your cluster design, this might place demands on your cluster database.
- **Install dependencies –** the installed version of SLD is for SAP NetWeaver 7.0 (2004s) SPS07 or higher. The SLD to which you register must be the latest Common Information Model version (currently 1.6.30).
- **Set up the SLD –** verify that the SLD is available to SAP Mobile Server and configured to receive data. For more information, see the *Post-Installation Guide* and the *User Manual* for your SAP NetWeaver version on SDN: *http://www.sdn.sap.com/irj/sdn/nw-sld*.

- **Set connection properties –** collect and deliver to the platform administrator. Determine the connection values to the SLD server, including its host name, protocol (HTTP or HTTPS), port, and the SLD user account.
- **Encrypt any dependent certificates –** if SLD is configured to use HTTPS, ensure that the SLD server certificate is signed by a known certificate authority. Otherwise, you must manually import the certificate into the SAP Mobile Server (primary node only) certificate keystore.

See *SAP SLD Server Overview* in *System Administration*.

See also

- *SAP License Audit Requirements* on page 7

SAP License Audit Requirements

For SAP applications like OData SDK Applications, administrators can generate an XML file that contains usage audit data that is then sent to the SAP License Audit. The XML file, which is compatible with the License Audit infrastructure, includes counts of users using the applications currently deployed to SAP Mobile Server.

To use SAP Mobile Platform data with SAP License Audit determine how the data is used. Review the contents of *SAP License Audit and Application Data Overview* in *System Administration* to see how data is extracted and delivered to SAP.

See also

- *SAP SLD Interoperability Requirements* on page 6

Network Communications Requirements

Network and host provisioning must accommodate SAP Mobile Platform internal communications.

When SAP Mobile Platform runtime components are installed on more than one host, they depend on network connections for some inter-process communication. Configure the local network to allow all communications between SAP Mobile Platform runtime components.

When all SAP Mobile Platform runtime components are installed on a single host, they depend on regular IP communication on the primary network interface of the host.

See also

- *Assessing High Availability and Disaster Recovery Objectives* on page 3
- *Host Platform Requirements* on page 4
- *SAP® Ecosystem Interoperability Requirements* on page 6
- *EIS Requirements* on page 10
- *Authentication and Authorization Security Requirements* on page 11

- *Intrusion Detection and Protection Requirements* on page 12

SAP Mobile Platform Port Accommodation

Infrastructure provisioning must accommodate all ports required by SAP Mobile Platform runtime components.

To accommodate SAP Mobile Platform ports, you may need to:

- Configure personal firewalls or host-based intrusion protection (HIPS) to allow access to component ports. See *Handling Intrusion Detection/Prevention Software* in the *Security* guide.
- Configure SAP Mobile Platform servers to change port number assignments.
  - You can change default port assignments for SAP Mobile Platform server components and SAP Control Center during installation.
  - You cannot change assignments of SAP Mobile Platform reserved ports. For a complete list, see *Reserved Ports*.
  - You can change some ports during cluster installation.

  Note: If there is a conflict for port 2480 or 2481, SAP Mobile Server will not start, and you cannot use SAP Control Center to change those SAP Mobile Server ports. You must temporarily stop the service that uses the conflicting port, then start SAP Mobile Server so you can change the port assignment from SAP Control Center.
- Reserve ephemeral ports on SAP Mobile Platform hosts to prevent other processes from using them.

  Note: Even if the installer does not detect a conflict, the Windows operating system may later use additional ports in the 1024 – 65535 range. In that event, you may encounter intermittent problems starting SAP Mobile Platform services.

  See the Microsoft operating system documentation to learn how to reserve ephemeral ports.

See also

- *Requirements for Load Balancers* on page 9
- *SAP Mobile Server Ports* on page 109
- *Data Tier Ports* on page 110
- *SAP Control Center Ports* on page 110
- *Relay Server Ports* on page 111
- *Reserved Ports* on page 111
- *Other Ports* on page 112

Requirements for Load Balancers

Client and DCN guidelines for deploying a third-party load balancer that is connected directly to SAP Mobile Server ports.

Note: SAP does not recommend or endorse any specific third-party load balancer appliance, device, or software.

See also

- *SAP Mobile Platform Port Accommodation* on page 8

Requirements for Client Load Balancing

Client guidelines for deploying a third-party load balancer that is connected directly to SAP Mobile Server client ports.

The load balancer must:

- Balance connections at the TCP/IP level.
- Perform all load balancing on client connections to the SAP Mobile Server cluster, independent of any other network device.
- Connect directly to the client ports of each SAP Mobile Server in the cluster.
- Support client request routing (that is, back-end server affinity) for nonpersistent HTTP connections, for these way the load balancer dispatches requests, based on the HTTP header or cookie:

Load Balancer Dispatches...	HTTP Header (or Cookie)
All client requests to Relay Servers	`ias-rs-sessionid`
Replication based synchronization (RBS) client requests directly to SAP Mobile Platform	`ml-session-id`
Other client requests directly to SAP Mobile Platform	`X-SUP-SESSID` (cookie)

  For REST API applications, see also *Reference* topic in *Developer Guide: REST API Applications*, especially the *HTTP Headers and Cookies* subtopic.
- Map reverse proxy ports to SAP Mobile Server ports.
- If both new version clients (Sybase Unwired Platform 2.2.0 and later, SAP Mobile Platform 2.3 and later) and old version clients (Sybase Unwired Platform 2.1.x and earlier) are supported, set up Relay Servers between SAP Mobile Server and load balancers and route client requests based on `ias-rs-sessionid`.
- If both replication-based synchronization (RBS) clients and other types of clients are supported, set up Relay Servers between SAP Mobile Server and load balancers and route client requests based on `ias-rs-sessionid`.

Note: To allow header inspection, the load balancer must be a transport-layer security (TLS) endpoint on HTTPS connections from mobile clients.

Requirements for DCN Load Balancing

DCN guidelines for deploying a third-party load balancer that is connected directly to SAP Mobile Server DCN ports.

The load balancer must:

- Balance connections at the TCP/IP level.
- Perform all load balancing on EIS connections to the SAP Mobile Server cluster, independent of any other network device.
- Connect directly to the DCN port of each SAP Mobile Server in the cluster.

EIS Requirements

You may need to provision some enterprise information system (EIS) resources to enable SAP Mobile Platform to use data services provided by the EIS.

See also

- *Assessing High Availability and Disaster Recovery Objectives* on page 3
- *Host Platform Requirements* on page 4
- *SAP® Ecosystem Interoperability Requirements* on page 6
- *Network Communications Requirements* on page 7
- *Authentication and Authorization Security Requirements* on page 11
- *Intrusion Detection and Protection Requirements* on page 12

EIS Driver Requirements

SAP Mobile Platform includes drivers for SAP databases, such as SAP® Adaptive Server® Enterprise and SAP® SQL Anywhere®, and Web services. For nonSAP data sources, such as DB2, Oracle, or Microsoft SQL Server, you must install the appropriate drivers.

Depending on the type of enterprise information system (EIS) connection, you may need to copy some driver and library files to the SAP Mobile Server installation directories.

In an SAP Mobile Server cluster, each host must have the appropriate drivers installed.

See *Supported Hardware and Software* for the most current supported versions of different EISes, and different versions of drivers for the same EIS.

See also

- *SAP External Libraries Requirements* on page 11

SAP External Libraries Requirements

Understand the requirements for external files you can optionally download from SAP and install into SAP Mobile Platform to enable communication with an SAP EIS.

- **SAP Cryptographic Libraries –** required by SAP Mobile Platform to enable Secure Network Communications (SNC) between SAP Mobile Server or SAP Mobile WorkSpace and the SAP EIS.
- **SAPCAR utility –** required to extract files from the SAP cryptographic library.

See also

- *EIS Driver Requirements* on page 10

Authentication and Authorization Security Requirements

If you do not effectively define security requirements in advance, you cannot evaluate the resulting system for success or failure prior to implementation. The features that currently exist determine how SAP Mobile Platform security is affected by your infrastructure.

- **Roles and distribution of assignments –** you can map roles at various levels: domain, security configuration, application, and package. To prevent mapping collisions, identify the roles that need to exist, and how to map them.
- **Security provider strategy –** identify the existing built-in security providers, and the ones you can create using the CSI API. The security providers you configure in SAP Mobile Platform pass authentication and authorization information to the provider used in your environment. Identifying the providers simplifies the implementation of a security configuration by the platform administrator after installation. If you are using SSO with a security provider, you may also need to prepare libraries and other back-end components. See the postinstallation requirements documented in *Stage 3: Implement*.

See also

- *Assessing High Availability and Disaster Recovery Objectives* on page 3
- *Host Platform Requirements* on page 4
- *SAP® Ecosystem Interoperability Requirements* on page 6
- *Network Communications Requirements* on page 7
- *EIS Requirements* on page 10
- *Intrusion Detection and Protection Requirements* on page 12

Intrusion Detection and Protection Requirements

To accommodate SAP Mobile Platform internal communications, you may need to reconfigure hardware or software intrusion detection/prevention systems.

- Configure "personal firewall" applications, or host-based intrusion prevention software (HIPS) to allow all communications between SAP Mobile Platform server components.
- To prevent required internal component communication ports from being blocked, configure an intrusion prevention system (IPS), or intrusion detection and prevention system (IDPS) appliances to allow connections to the ports SAP Mobile Platform uses.
- When you install any new intrusion detection/prevention system on an SAP Mobile Platform server host, or on a local network that services an SAP Mobile Platform server host, configure that new system to accommodate all SAP Mobile Platform internal communications.

See also

- *Assessing High Availability and Disaster Recovery Objectives* on page 3
- *Host Platform Requirements* on page 4
- *SAP® Ecosystem Interoperability Requirements* on page 6
- *Network Communications Requirements* on page 7
- *EIS Requirements* on page 10
- *Authentication and Authorization Security Requirements* on page 11

Stage 2: Design

In the design stage, you create the landscapes, process diagrams (beyond the scope of this document), perform license selection, and prepare supporting documentation (including installation worksheets). The design stage uses the requirements identified during the assessment stage.

Each requirement produces a set of one or more design elements which are intended to describe the software in sufficient detail allow any IT member to perform the installation.

Understanding Landscape Options

Gathering information about all installable options for SAP Mobile Platform provides the required foundation for making implementation decisions for your installation environment.

Single-Server Installations

In a single-server installation, one SAP Mobile Server node and a data tier are installed during a single installation procedure, executed on a single host.

A single-server SAP Mobile Platform Runtime system is simpler, less expensive to deploy, and generally easier to maintain. However, it also has significant limitations:

- It cannot be scaled by adding or subtracting servers, to adapt to changes in system load or performance requirements.
- It cannot take advantage of conventional load-balancing and failover mechanisms to provide a greater level of system availability.
- The only way you can increase overall system performance is by upgrading the host system resources (CPU, RAM, and so on).

Note: You cannot upgrade a nonclustered SAP Mobile Platform system to a clustered system. You must redeploy the SAP Mobile Platform system, using cluster installation options on suitable hosts.

Plan carefully when you choose between clustered and nonclustered designs. If you can foresee any future requirement for a clustered system, such as a service level agreement (SLA) that would require scalability, or higher system availability, consider initially deploying a clustered SAP Mobile Platform system.

Cluster Installations

In a cluster installation, data tiers, SAP Mobile Server nodes that are installed as application server nodes, and (optional) SAP Mobile Server scale-out nodes, are installed by separate installation procedures, typically on separate hosts.

There are two main advantages of a clustered SAP Mobile Platform system:

- It can be scaled by adding or subtracting servers (nodes in a cluster), to adapt to changes in system load or performance requirements.
- Redundant cluster nodes allow conventional load-balancing and failover mechanisms to provide a greater level of system availability.

In a typical clustered system, SAP Mobile Server instances do not share host system resources with data tier servers.

Choose a clustered system design to meet requirements for scalability, higher availability, and overall higher system performance.

SAP Mobile Server Load-Balancing Clusters

The SAP Mobile Server cluster enables load balancing to improve system availability and performance.

An SAP Mobile Server cluster consists of two or more SAP Mobile Servers that:

- Service the same set of client devices, users, and mobile applications
- Rely on the same set of enterprise information systems to provide back-end data services
- Rely on the same data tier resources to provide runtime data services

Because they share common data tier resources, all SAP Mobile Servers in the cluster have access to the same cached data from the EIS, messaging data for clients, cluster and server configuration data, and system log data.

The common data tier lets you easily scale the SAP Mobile Server cluster, adding or removing nodes at any time.

With a load balancer, such as Relay Server, or a third-party load balancer appliance:

- SAP Mobile Servers in the cluster can share workloads, improving performance and efficiency
- Clients have a common point of access, independent of any particular SAP Mobile Server in the cluster

Data Tier Failover Clusters

The data tier failover cluster provides high availability and fault tolerance.

Note: The data tier cluster relies on cluster services provided by a Windows Server operating system (such as Microsoft Cluster Service, or Failover Clustering).

A data tier cluster consists of:

- Two data tier hosts, each managed by a Windows-based failover cluster service
- At least one fault-tolerant storage device, which provides the shared cluster storage for database files and transaction logs

Each data tier host is a redundant node in the failover cluster—one active, and one standby (or passive). Host system performance is more critical for the data tier servers, because each host must assume the entire load imposed by the SAP Mobile Server cluster.

To deploy the data tier in a failover cluster:

- All data tier server software must be installed on a local drive, on each data tier host. The data tier software cannot be installed on shared cluster storage, or on any storage resource that can be managed independent of the data tier host.
- All data tier database files and transaction logs must be located on a shared-cluster storage device, such as a SAN device that is assigned the appropriate RAID level.
- Each data tier host must be physically connected, by a host bus adapter, to the shared cluster storage device. Each volume that houses a database file system must be accessed as a local disk, on each data tier host.
- All data tier services must be configured as cluster resources, managed in the context of a common cluster instance.

To a SAP Mobile Server, data tier hosts in a failover cluster appear to be a single, logical data tier entity.

Note: Follow current Microsoft guidelines for networks, hosts, and storage devices used in a failover cluster.

Enhanced Load Balancing

A cluster of two or more SAP Mobile Server nodes provides some degree of load balancing, but load balancing can also be applied to both client connections and EIS connections, to improve performance and efficiently use resources in the SAP Mobile Server cluster.

There are two load-balancing mechanisms you can use with the SAP Mobile Server cluster:

- Relay Server – an SAP software product that acts as a reverse proxy server for client devices communicating with SAP Mobile Server.
- Load balancer appliance – a third-party product, such as an L4 network switch, can be used for both client and EIS connections.

Note: You cannot use Relay Server on EIS connections.

Client Load Balancing

Client load balancing improves capacity and performance of the SAP Mobile Server cluster when it is servicing mobile client requests.

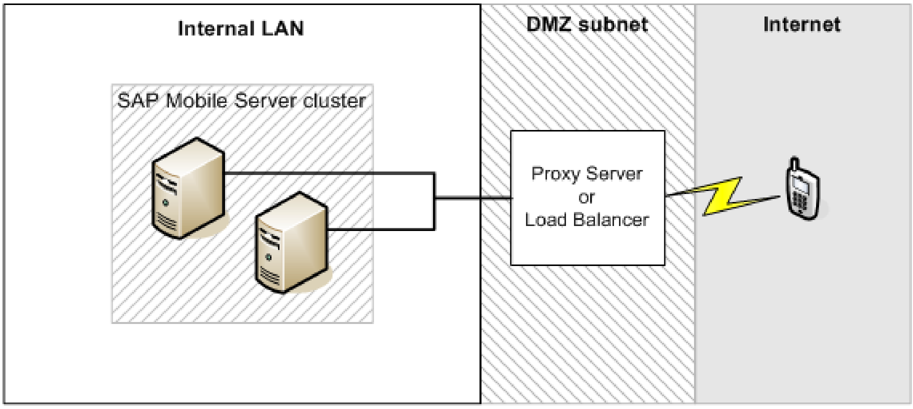
With load balancing on client connections:

- SAP Mobile Servers in the cluster can share the client workload, improving the efficiency of services to client devices.
- Clients have a common point of access, independent of any particular SAP Mobile Server instance.

There are two types of client load balancing to consider:

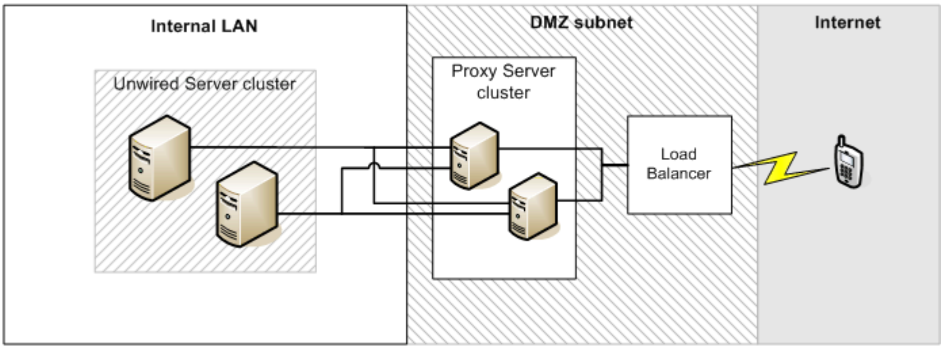
- **Simple load balancing –** you can deploy either one proxy server, or a load balancer appliance, to implement client load balancing on a single network node.

Figure 1: Load Balancing with One Proxy Server or Load Balancer



- **Clustered load balancing –** for higher capacity or higher availability, you can deploy a proxy server cluster, with or without a front-end load balancer appliance.

Figure 2: Load Balancing with Proxy Server Cluster



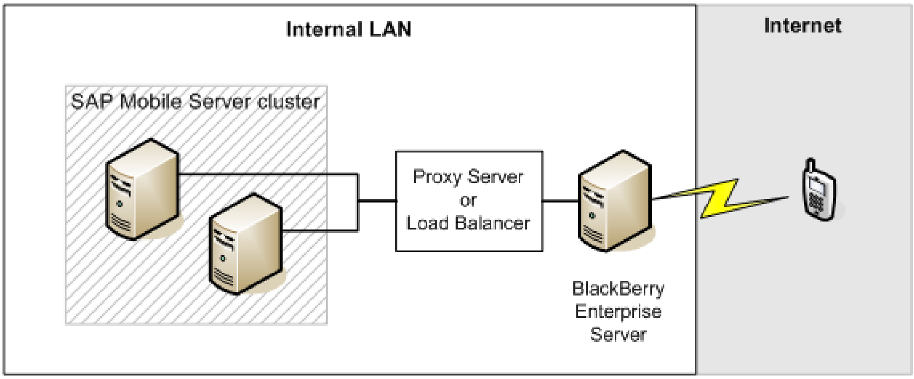
Client Load Balancing with BES

You can apply load balancing to connections between BlackBerry Enterprise Server (BES) and SAP Mobile Servers in a cluster.

When you deploy an SAP Mobile Platform system to support BlackBerry device users, BES treats the SAP Mobile Server as a back-end enterprise application. Connections from BES are treated as client connections by SAP Mobile Server.

To implement client load balancing on connections between BES and SAP Mobile Servers in a cluster, you can deploy either a proxy server, or a load balancer appliance.

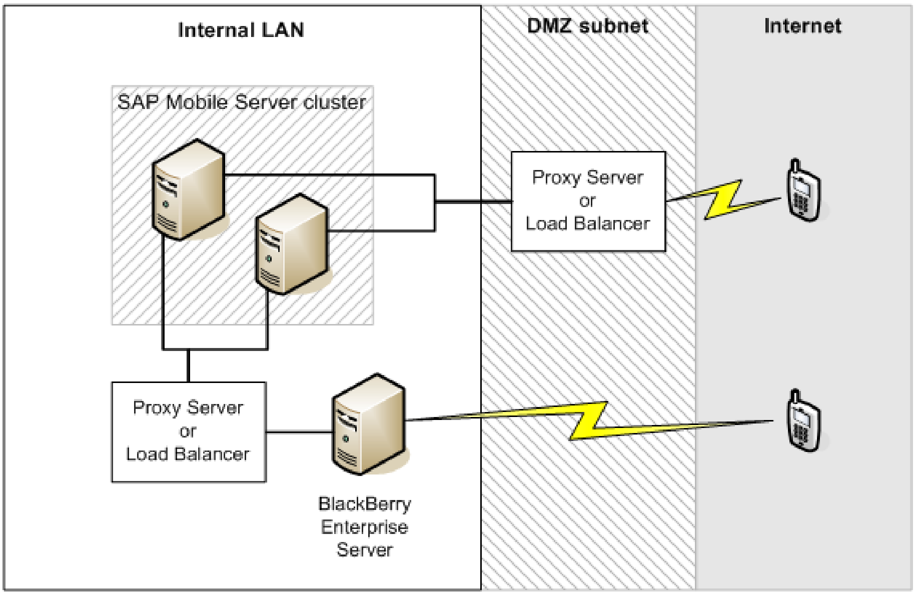
Figure 3: Load Balancing with BES (BlackBerry Clients)



The load-balancing mechanism between BES and SAP Mobile Server is deployed on the internal LAN.

If your SAP Mobile Platform system supports both BlackBerry device users and users of other device types, you can apply load balancing to both BES connections and other client connections.

Figure 4: Load Balancing with Both BlackBerry and Other Clients


Internal LAN
DMZ subnet
Internet
SAP Mobile Server cluster
Proxy Server or Load Balancer
Proxy Server or Load Balancer
BlackBerry Enterprise Server

EIS Load Balancing

EIS load balancing improves the capacity and performance of the SAP Mobile Server cluster when it services data change notification (DCN) or SAP® Data Orchestration Engine Connector requests from the back-end enterprise information system (EIS).

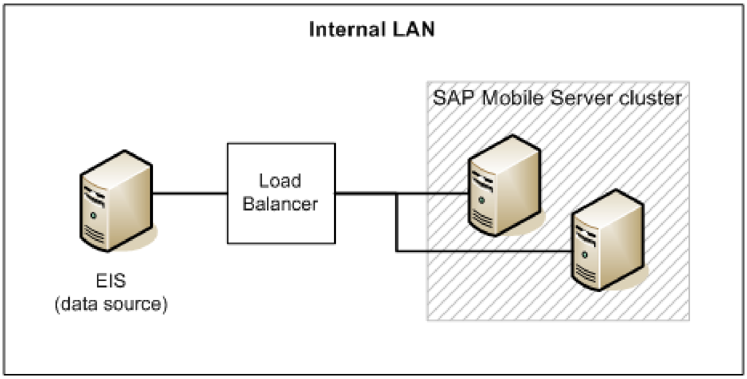
With load balancing on EIS connections:

- SAP Mobile Servers in the cluster can share DCN or DOE-C workloads, improving the efficiency of service for mobile applications that rely on replication synchronization (or data that is “pushed” from the server).
- The EIS need not rely on a connection to any particular SAP Mobile Server in the cluster, eliminating the SAP Mobile Server as a single point of failure in DCN or DOE-C processing.

To implement load balancing on connections between the EIS and SAP Mobile Server, you must use a third-party load balancer.

You cannot use proxy server on connections between the EIS and SAP Mobile Server.

Figure 5: Load Balancing on EIS Connection



Selecting a Landscape Design Based on High Availability and Disaster Recovery Requirements

Based on your assessment of high availability and disaster recovery objectives, as well as factoring in other considerations that affect your implementation of SAP Mobile Platform, you can select one of several common landscape designs to use as a guideline.

The landscape designs detailed in this section are common designs that provide some guidelines for typical SAP Mobile Platform implementations. There are many possible landscape designs, and you should rely on solution architects who are familiar with all the requirements and specific circumstances to recommend a satisfactory approach for your enterprise.

1. Determine your parameters for recovery time objective (RTO) for downtime and recovery point objective (RPO) for data loss.

   See *Assessing High Availability and Disaster Recovery Objectives* in *Stage 1: Assess* in this document.

2. Select the landscape design that best matches your objectives for downtime and data loss.

   You may need to evaluate several of these landscapes to make your final selection.

Downtime/Data Loss Objectives*	Landscape Design
Downtime: user's best effort determines downtime Data Loss: complete data loss is tolerable	*Individual Development and Test* on page 20
Downtime: 4 – 24 hours Data Loss: 15 minutes – 24 hours	*Shared Development and Test* on page 21

Downtime/Data Loss Objectives*	Landscape Design
Downtime: less than 4 hours Data Loss: no data loss	*Fault-Tolerant Production* on page 22

* The values for downtime and data loss are guidelines, based on sample installations that are operating in relatively isolated environments. In more complex environments, actual values for these parameters, especially for downtime, are typically adversely affected by the need to simultaneously restore other systems.

Individual Development and Test

For an individual working autonomously to develop or test applications, a single-server installation is generally sufficient. This landscape design does not include high availability or a disaster recovery infrastructure.

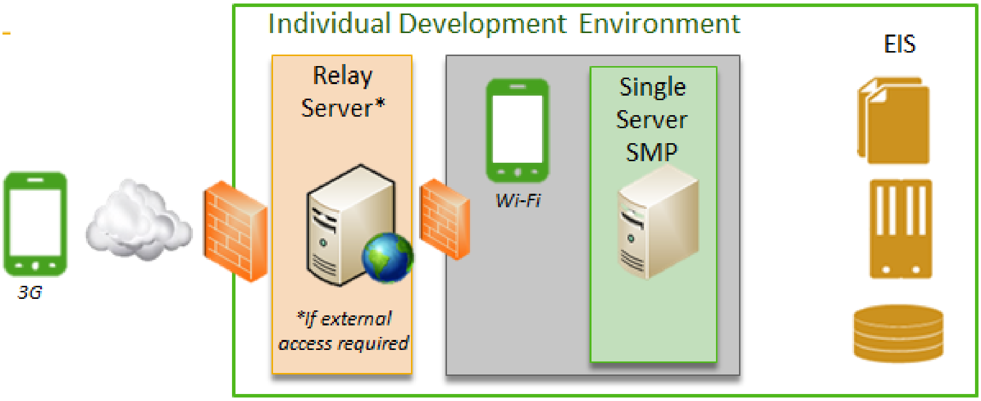
High Availability and Disaster Recovery Parameters

- **Downtime:** user's best effort determines downtime
- **Data Loss:** complete data loss is tolerable

Typical Use Cases

- Individual development environment

Landscape Design


Individual Development Environment
Relay Server*
*If external access required
3G
Wi-Fi
Single Server SMP
EIS

- Single-server environment
- No special backup infrastructure
- Relay Server in DMZ if external access is required

Installation Scenario to Implement

- *Single-Server Installation* on page 24

Shared Development and Test

A basic cluster installation supports a team working together to develop or test applications. To minimize work disruption, this landscape design includes a minimal high availability and disaster recovery infrastructure.

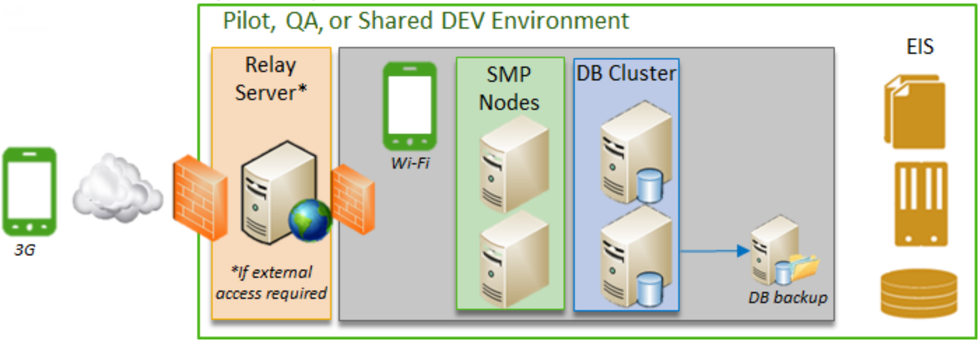
High Availability and Disaster Recovery Parameters

- **Downtime:** 4 – 24 hours
- **Data Loss:** 15 minutes – 24 hours

Typical Use Cases

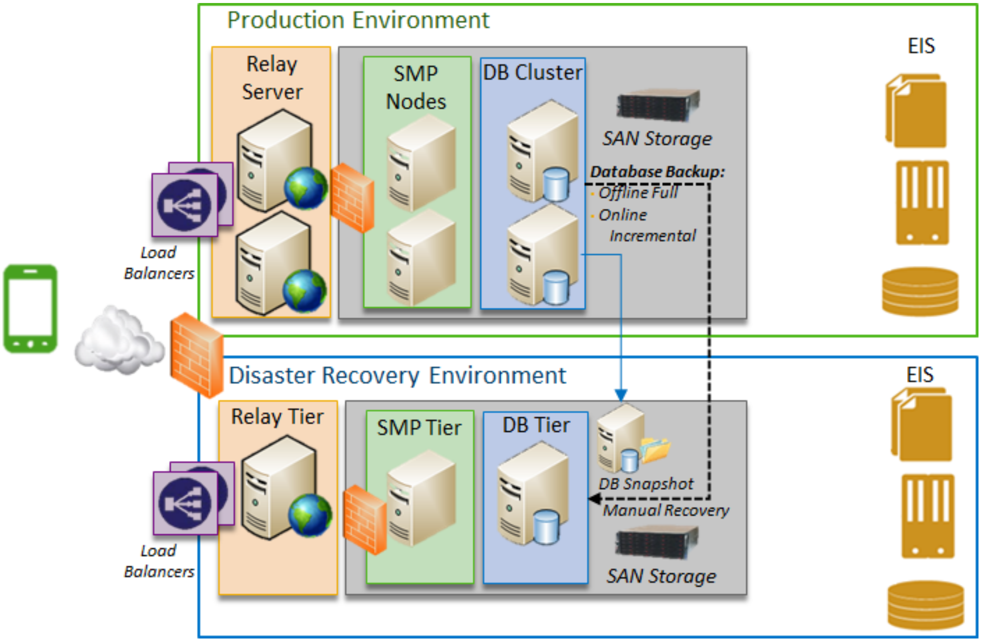
- Enterprise productivity tools
- QA, prototype, proof-of-concept, or pilot environments
- Shared or complex development environments
- Deprecated, offline business applications that are retained for historical purposes

Minimum Landscape Design


Pilot, QA, or Shared DEV Environment
Relay Server*
*If external access required
Wi-Fi
SMP Nodes
DB Cluster
DB backup
EIS
3G

- Clustered environment
- On-demand offline full backup, based on RPO target
- No disaster recovery environment
- Manual recovery process

Enhanced Landscape Design



- Clustered environment
- Redundancy in all tiers; no single point of failure
- Offline full backup (ad hoc)
- Online incremental backup
- Database snapshots
- Manual recovery process

Installation Scenarios to Implement

- *Simple Load-Balancing Cluster* on page 27, installed as both production and recovery systems

Fault-Tolerant Production

A cluster design in which a production system is mirrored by a complete disaster recovery system supports production applications that require no data loss and minimal downtime.

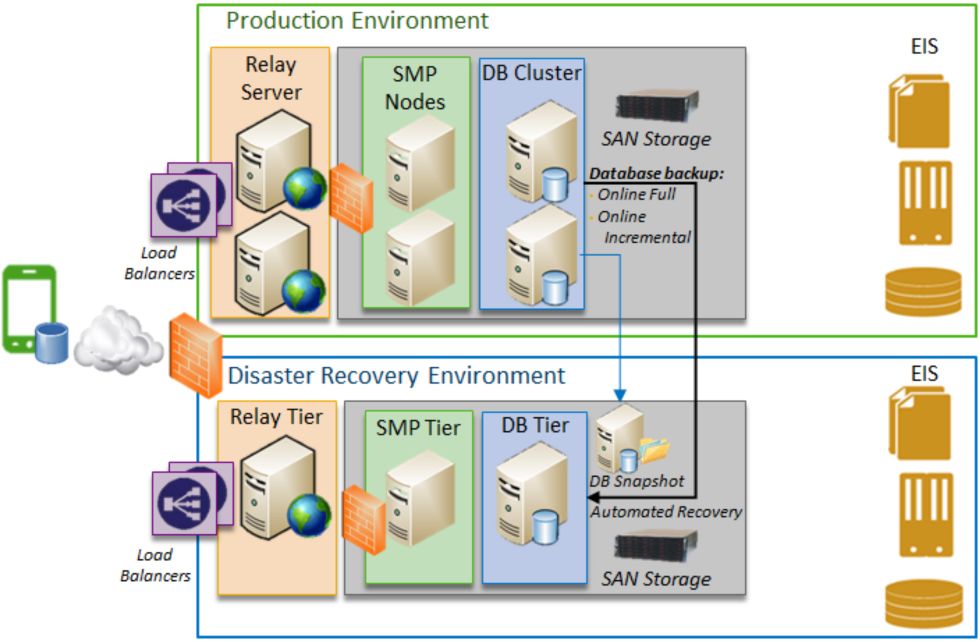
High Availability and Disaster Recovery Parameters

- **Downtime:** Minimal-to-zero
- **Data Loss:** No data loss

Typical Use Cases

- Mission-critical enterprise applications
- Customer-facing systems

Landscape Design



- Clustered environment
- Redundancy in all tiers; no single point of failure
- Environment replica in separate disaster recovery environment
- Online full backup
- Online incremental backup
- Online synchronous mirroring
- Client database backup
- Frequent database snapshots
- Rolling maintenance
- Active system health monitoring

Installation Scenarios to Implement

- *Simple Load-Balancing Cluster* on page 27, installed as both production and recovery systems, or,

- *Standard Microsoft Failover Cluster* on page 29, installed as both production and recovery systems

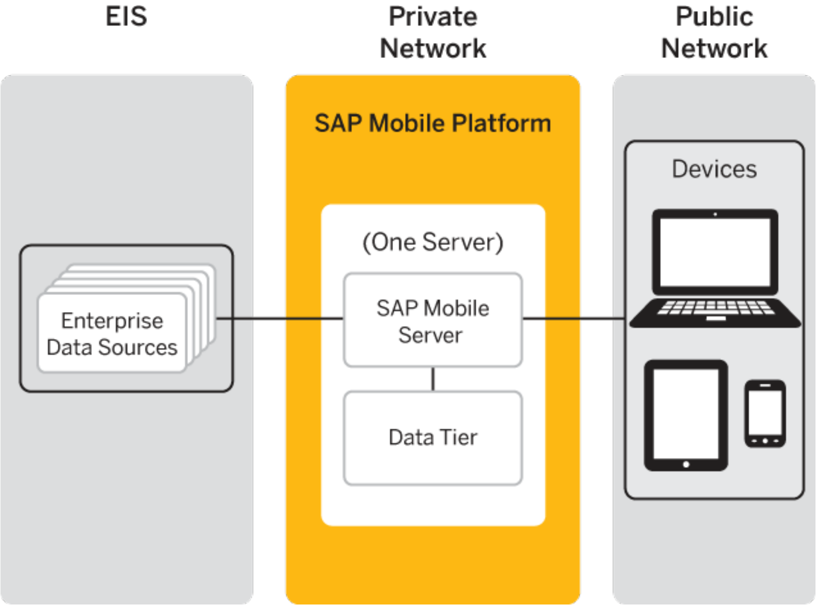
Choosing Installation Scenarios to Implement Your Landscape Design

Each SAP Mobile Platform installation scenario outlines a specific way to install and configure an SAP Mobile Platform system. Your landscape design determines the installation scenario you will implement.

Single-Server Installation

Single-server installations set up a complete SAP Mobile Platform system on a single server. This configuration is suitable for a low-volume production system where continuous uptime is not critical. With SAP Mobile SDK installed on the same system, a single-server installation provides a self-contained developer workstation.

Figure 6: Single-Server Installation



Summary of single-server installation architecture:

- Consists of a data tier and a single SAP Mobile Server node.
- The SAP Mobile Server and the data tier are installed on the same server.

- No proxy servers are needed; load balancing is not an issue with a single SAP Mobile Server instance.

Table 1. Design Characteristics and Use Cases

Consideration	Details
Recommended for	• Short-term use: trial or evaluation. • Long-term use: single-user development and testing. • Long-term use: development by a small team with light data loads.
Not recommended for	• Enterprise development by a large team. • Production deployment.
Worksheet to complete	*Install Single Server* worksheet. See *Completing Installation Worksheets* on page 39.
Installation procedure used	*Installing SAP Mobile Platform on a Single Server* in *Installation Guide for Runtime*.
Limitations	• Not easily scalable. • No load-balancing or failover support. • Host systems must be adequate to support all of the applications and services included in all SAP Mobile Platform server components. • Disk resources must be adequate to support all databases managed by the data tier.
Benefits	• Low administration complexity. • Easily secured. • No dependency on network connections between the platform server components.

See also

- *Simple Load-Balancing Cluster* on page 27
- *Standard Microsoft Failover Cluster* on page 29
- *Microsoft Failover Cluster with Shared Hosts* on page 31

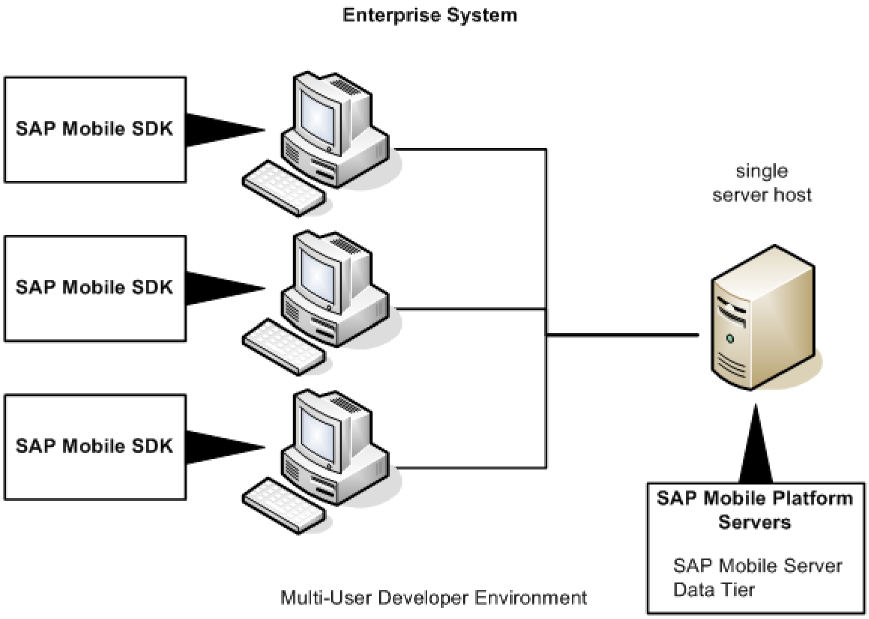
Single-Server System Designs

Single-node designs typically install all SAP Mobile Platform server components (the SAP Mobile Server and data tier servers) on a single host, using a single installation procedure.

There are two scenarios often selected:

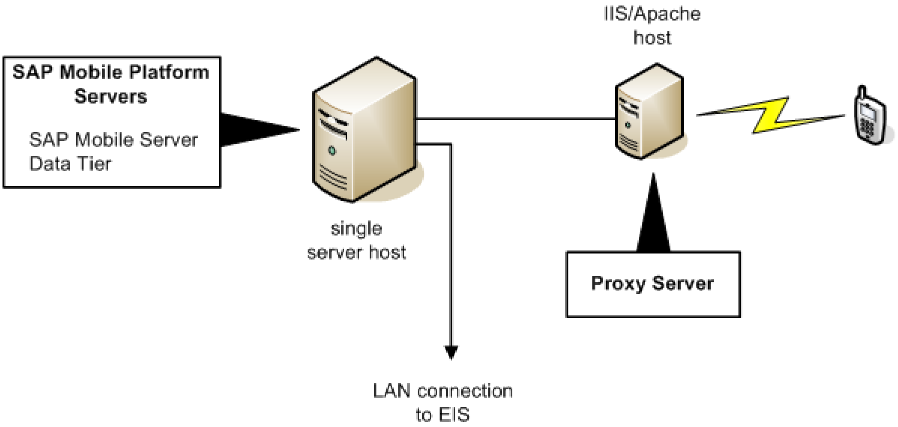
- **Enterprise development –** an enterprise system scenario that supports a developer team environment in which multiple developer workstations share a common SAP Mobile Platform Runtime installation.

Figure 7: Single-Node Enterprise System (Developer Team)



- **Online Data Proxy support environments –** SAP Mobile Platform can support an Online Data Proxy production system.

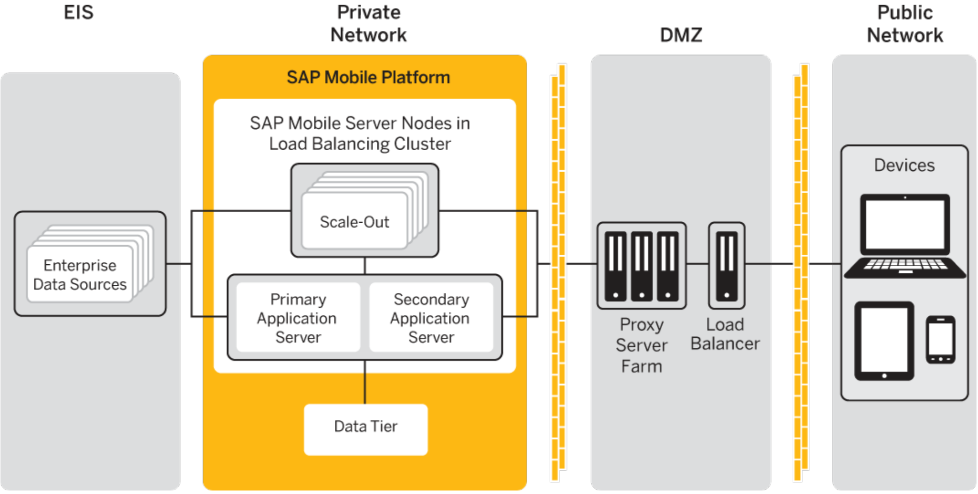
Figure 8: Single-Node Enterprise System (Online Data Proxy)



Simple Load-Balancing Cluster

A simple load-balancing cluster installs SAP Mobile Platform Runtime components onto multiple servers that are configured to share the workload, with additional (optional) SAP Mobile Server nodes installed as scale-out nodes. With two SAP Mobile Server application server nodes, the system continues to operate if either application server node fails.

Figure 9: Simple Load-Balancing Cluster



Summary of simple load-balancing cluster architecture:

- Consists of two SAP Mobile Servers, installed as application server nodes, and as many optional SAP Mobile Server scale-out nodes as needed. If the primary SAP Mobile Server application server node fails, the secondary application server node takes over and application server support continues uninterrupted.

Note: Scale-out nodes take requests only from messaging clients (OData SDK, Hybrid Web Container) and HTTP clients (REST APIs). For these clients to connect to the scale-out node, clients must be built with SAP Mobile Platform version 2.3 or later, or Unwired Platform 2.2 SP01 or later. Only clients from SAP Mobile Platform version 2.3 or later, and Unwired Platform 2.2 SP01 or later, can fully support HTTP cookies. You must migrate existing clients to version 2.3 to connect to scale-out nodes. For details about cookie support, see the corresponding *Developer Guide* for your client type.

- A single data tier node supports the SAP Mobile Server nodes.
- Each SAP Mobile Server node and the data tier node are installed on separate servers.
- A proxy server farm provides enhanced load balancing for distributing requests to the different SAP Mobile Server nodes.
- A hardware load balancer optimizes the performance of the proxy server farm.

Table 2. Design Characteristics and Use Cases

Consideration	Details
Recommended for	• Multiuser development and test environments of smaller scale • Low-volume production environments, to maintain performance requirements while adapting to changing demands
Degree of collocation	None or partial, depending on hardware resource restrictions.
Not recommended for	• High-volume production environments
Worksheet to complete	*Install Simple Load-Bal Cluster* worksheet. See *Completing Installation Worksheets* on page 39.
Installation procedure used	*Installing SAP Mobile Server in a Simple Load-Balancing Cluster* in *Installation Guide for Runtime*
Limitations	• No failover support for the database; single point of failure risk • Disk resources must be adequate to support all databases managed by the data tier • Increased complexity of managing and monitoring the server hardware and its associated software

Consideration	Details
Benefits	• Scalable • Improved performance

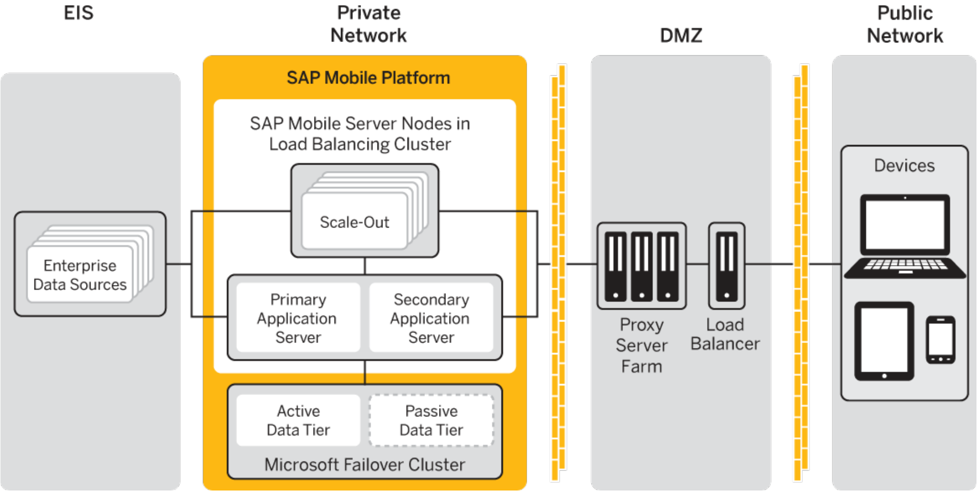
See also

- *Single-Server Installation* on page 24
- *Standard Microsoft Failover Cluster* on page 29
- *Microsoft Failover Cluster with Shared Hosts* on page 31

Standard Microsoft Failover Cluster

Installing two data tier nodes in a Microsoft Failover Cluster ensures that the system can continue to operate if the active data tier node fails. Adding this capability to the load-balancing cluster for SAP Mobile Server nodes ensures there is no single point of failure in the system.

Figure 10: Standard Microsoft Failover Cluster



Summary of standard Microsoft Failover Cluster architecture:

- Consists of two data tiers installed in a Microsoft Failover Cluster support the SAP Mobile Server nodes. If the active data tier node fails, the passive data tier node becomes active and database support continues uninterrupted.
- Each SAP Mobile Server node and the two data tier nodes are installed on separate servers.
- A proxy server farm provides enhanced load balancing for distributing requests to the different SAP Mobile Server nodes.

- A hardware load balancer optimizes the performance of the proxy server farm.

Table 3. Design Characteristics and Use Cases

Consideration	Details
Recommended for	• Multiuser development and test environments of larger scale. • High-volume production environments that require some combination of scalability, high availability, and high performance.
Degree of collocation	None, full distribution.
Not recommended for	Small-scale environments, or organizations with budget concerns/restrictions.
Worksheet to complete	*Install MS Failover Cluster* worksheet. See *Completing Installation Worksheets* on page 39.
Installation procedure used	*Installing SAP Mobile Platform with a Standard Microsoft Failover Cluster* in *Installation Guide for Runtime*.
Limitations	• Disk resources must be adequate to support all databases managed by the data tier servers. • Increased complexity of managing and monitoring the server hardware and its associated software. • Most expensive due to added infrastructure costs associated with complete distribution. • Increased response times due to the increased load on the secondary server or the need to synchronize state information on or from multiple servers.
Benefits	• Improved performance. • No single point of failure. • Easily accommodates routine maintenance and upgrading. Because data tier is a failover cluster, this design allows for system downtime without affecting availability.

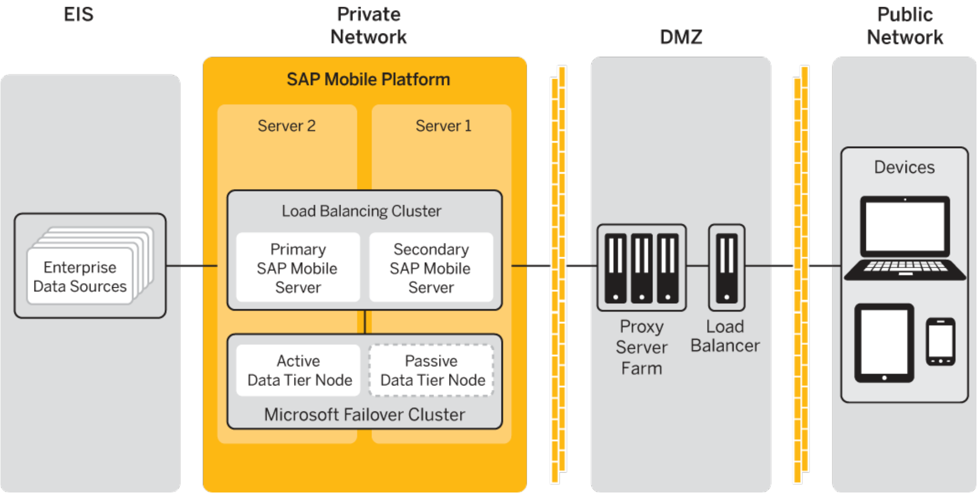
See also

- *Single-Server Installation* on page 24
- *Simple Load-Balancing Cluster* on page 27
- *Microsoft Failover Cluster with Shared Hosts* on page 31

Microsoft Failover Cluster with Shared Hosts

Placing one data tier and one SAP Mobile Server node, installed as an application server, on each of two physical servers, is a modification of the standard Microsoft Failover Cluster. This is the minimum hardware configuration that ensures there is no single point of failure in the system.

Figure 11: Microsoft Failover Cluster on Shared Hosts



Summary of Microsoft Failover Cluster architecture on shared hosts:

- Resembles two single-server installations, each with an SAP Mobile Server node and a data tier installed on a single server. However, the data tiers are also incorporated in a Microsoft Failover Cluster, so that if the active data tier node fails, the passive data tier node becomes active and database support continues uninterrupted.
- Consists of two data tiers installed in a Microsoft Failover Cluster support the SAP Mobile Server nodes. If the active data tier node fails, the passive data tier node becomes active and database support continues uninterrupted.
- A proxy server farm can optionally provide enhanced load balancing for distributing requests to the two SAP Mobile Server nodes, but if you select the shared host configuration because of cost considerations, you may also want to omit proxy servers.
- A hardware load balancer can optimizes the performance of a proxy server farm, if one is present.

Table 4. Design Characteristics and Use Cases

Consideration	Details
Recommended for	• Multiuser development and test environments of larger scale, where economizing on hardware costs is of prime importance. • Moderate-volume production environments that require some high availability.
Degree of collocation	Maximum collocation possible without a single point of failure.
Not recommended for	High-volume production environments.
Worksheet to complete	*Install MS Failover Cluster* worksheet. See *Completing Installation Worksheets* on page 39.
Installation procedure used	*Installing SAP Mobile Server with a Microsoft Failover Cluster with Shared Hosts* in *Installation Guide for Runtime*.
Limitations	• Host systems must be adequate to support collocation (if used). • Disk resources must be adequate to support all databases managed by the data tier servers. • Increased complexity of managing and monitoring the server hardware and its associated software. • Higher infrastructure costs. • Increased response times due to the increased load on the secondary server or the need to synchronize state information on or from multiple servers.
Benefits	• Improved performance. • No single point of failure. • Easily accommodates routine maintenance and upgrading. Because data tier is in a failover cluster, this design allows for system downtime without affecting availability.

See also

- *Single-Server Installation* on page 24
- *Simple Load-Balancing Cluster* on page 27
- *Standard Microsoft Failover Cluster* on page 29

Choosing Licenses

Before installing SAP Mobile Platform, determine your license type.

For evaluation versions of SAP Mobile Platform, you do not need a license to perform a single-server installation (all components installed to a single host). For all other installations, the appropriate license must be loaded and available on all hosts before you can run the installer.

1. *Assessing License Needs*

   Identify your environment type, and the license model you need to support. These criteria help you choose the license type so you can purchase licenses.
2. *Mapping Environment to Product Editions and License Types*

   Once you have identified your environment type and the system design you need to support, map those requirements to the available license types.
3. *Purchasing Licenses Before Installing*

   Once you have identified your product edition and license type, you can proceed with purchasing the licenses.
4. *License Validation*

   Attributes in a license file define the number of SAP Mobile Server instances that are allowed to run concurrently and the license expiration date.

Assessing License Needs

Identify your environment type, and the license model you need to support. These criteria help you choose the license type so you can purchase licenses.

Two attributes of each SAP Mobile Platform server license determine your license requirements:

- **Product edition** – addresses the SAP Mobile Platform system design options and your intended use.
- **License type** – addresses other license terms, such as per-seat (workstation) or per-core (server) allowances, and number of mobile devices, users, or applications supported.

See also

- *Mapping Environment to Product Editions and License Types* on page 36

Environments and Product Editions

You can deploy SAP Mobile Platform to different environments. SAP Mobile SDK (development) and SAP Mobile Platform Runtime components are licensed separately. The type of environment targeted helps you to determine the appropriate product edition and license type.

- Development is a preproduction environment where applications are developed on a single host, using a Personal Development Server license. A simple load-balancing cluster may be necessary, using an Enterprise Development Server license, if application performance is to be tested. For this environment, purchase a SAP Mobile SDK license for each developer workstation.
- Qualification is another preproduction environment that is used to test applications and runtime properties. If your budget allows, qualification environments should replicate production environments as closely as possible. A Microsoft Failover Cluster is used, with SAP Mobile Platform installed using an Enterprise Development Server license. Any developer workstation involved requires its own SAP Mobile SDK license.
- Production is a live runtime environment that uses a Microsoft Failover Cluster, with SAP Mobile Platform installed using an Enterprise Server license. For this environment, you require SAP Mobile Platform Runtime that can be licensed according to the number of CPU cores and developer workstations used. If any developer workstations are involved, each requires its own SAP Mobile SDK license.

Server Product Editions

SAP Mobile Platform Runtime components are licensed according to the product edition.

All SAP Mobile Platform Runtime editions include SAP Mobile Server and data tier components.

Do not use SAP Mobile Platform Runtime licensed under development-specific product editions (Personal Development Server and Enterprise Development Server) in a production system.

Product Edition	Summary
Personal Development Server PE Code = PD	• Allows use in development systems and testing systems only; not for use in production systems. • Requires all SAP Mobile Platform server components to be installed on the same, single-user host with SAP Mobile SDK.
Enterprise Development Server PE Code = ED	• Allows use in development systems and testing systems only; not for use in production systems. • Allows each installable component to be located on a separate host. • Allows clustered systems.

Product Edition	Summary
Enterprise Server PE Code = EE	• License type determines allowed use (production only, or development and testing only). • Allows each installable component to be located on a separate host. • Allows clustered systems.

Note: You can install the SAP Mobile Platform Enterprise Server Edition only on 64-bit operating systems. You can install the Personal Development Server and Enterprise Development Server Editions on either 32-bit or 64-bit operating systems.

License Types

Each license type is associated with one or more SAP Mobile Platform product editions.

License Type	Summary
Standalone seat license LT Code = SS	• SAP Mobile Platform Runtime components must be installed on same host as SAP Mobile SDK. • Available only with Personal Development Server Edition.
Development and test license LT Code = DT	• Servers licensed for development and testing use only; no production use allowed. • No limit on CPU/cores or host configuration (single or multiple hosts, clusters, and so on) • Available with Enterprise Development Server and Enterprise Server Editions.
CPU/core license LT Code = CP	• Servers licensed by CPU/cores, for production use only; no development and testing use allowed. • No limit on host configuration (single or multiple hosts, clusters, and so on). • Available only with Enterprise Server Edition.
OEM license LT Code = AS	• SAP Mobile Platform servers to be bundled with packaged applications and redistributed. • Servers licensed for production use only; no development and testing use allowed. • Unserved license only, same license on all server hosts (no host ID required), no limit on CPU/cores or host configuration (single or multiple hosts, clusters, and so on). • Clients only are counted for licensing/royalties (various terms). • Available only with Enterprise Server edition.

License Deployment Models

Software licenses for SAP Mobile Platform use the Sybase® Software Asset Management (SySAM) system. SySAM provides two license deployment models from which to choose.

- **Unserved license –** each license is granted for one specific host. The license file must be stored locally, on the SAP Mobile Platform host. The license cannot be transferred to another host.
- **Served license –** a license is granted for a number of hosts. The license file is stored on a SySAM license server, and the license can be automatically acquired (checked out) by any SAP Mobile Platform host.

If you choose the served license model, you must deploy a SySAM license server to support the system, and you must enable network communications between the SySAM license server and all SAP Mobile Platform Runtime hosts.

Refer to the *SySAM Users Guide* for details.

Mapping Environment to Product Editions and License Types

Once you have identified your environment type and the system design you need to support, map those requirements to the available license types.

1. Choose a product edition for the environment into which SAP Mobile Platform is being installed:

Environment	Product Editions Supported
Development	ED, PD
Qualification	ED (pilot) or EE (pilot/QA testing)
Production	EE and ED

2. Choose a license type for the product edition selected:

	SS	DT	CP
PD	X		
ED		X	
EE		X	X

See also

- *Assessing License Needs* on page 33
- *Purchasing Licenses Before Installing* on page 37

Decision Criteria: DT and CP Licenses for EE Servers

The Enterprise Server Edition is available with DT and CP licenses.

If you are choosing between these license types, review these criteria to help you evaluate which option to choose.

Criteria	CP	DT
Usage	Production	Testing
Server license base	Per core	Per core
SAP Mobile WorkSpace	Requires separate product edition, separate license	Requires separate product edition, separate license

Purchasing Licenses Before Installing

Once you have identified your product edition and license type, you can proceed with purchasing the licenses.

1. **1.** Review your landscape design, and determine the number of nodes in your deployment.

   This determines the number of licenses you need to purchase. Each redundant node is separately licensed.
2. **2.** Arrange for the purchase of the number of licenses requred.
3. **3.** After the purchase of licenses is completed, download the licenses.

   When you purchase SySAM 2-enabled SAP products, you must generate, download, and deploy SySAM product licenses.
   - If you ordered your product under an SAP® contract and were directed to download from SAP Service Marketplace (SMP), you can use SMP at *http://service.sap.com/licensekeys* (login required) to generate license keys for SAP products that use SySAM 2-based licenses.
   - If you purchased your product from Sybase® or an authorized Sybase reseller, go to the secure Sybase Product Download Center (SPDC) at *https://sybase.subscribenet.com* and log in to generate license keys. The license generation process may vary slightly, depending on whether you ordered directly from Sybase or from a Sybase reseller.

   For license download and installation instructions, see *Obtaining a License* in *Installation Guide for Runtime*.

See also

- *Mapping Environment to Product Editions and License Types* on page 36

License Validation

Attributes in a license file define the number of SAP Mobile Server instances that are allowed to run concurrently and the license expiration date.

Each SAP Mobile Server instance must have its own server license.

Note: In a clustered design, choose the served license deployment model to enable license coordination in the SAP Mobile Server cluster.

The SAP Mobile Server checks the server license. At startup, if the SAP Mobile Server cannot retrieve the number of licensed servers from the license file, or if the server is not licensed, the SAP Mobile Server stops (or enters the license grace period, if any).

The SAP Mobile Server writes all license errors to the log.

Stage 3: Implement

The implementation stage uses the documentation produced by the design stage. IT personnel use any documentation, for example, completed worksheets design documents, or diagrams, as the concrete guidelines for executing and phasing the installation.

Completing Installation Worksheets

To streamline installation tasks, use the installation worksheet specific to your deployment scenario.

You may want to start using the worksheets during the design and planning stage. They might then be completed by the person performing the actual installation according to the design documentation you deliver.

Complete the worksheet for your chosen scenario. Obtain the Excel workbook file with the scenario worksheets by clicking: *../misc/SMP23_Worksheets.zip*.

See also

- *Performing the Installation* on page 39
- *Completing New and Upgrade Installations* on page 40
- *Adding Relay Servers or Reverse Proxies* on page 49
- *Agentry Server Host System Connectivity* on page 89
- *The Agentry Server in SAP Mobile Platform Clustered Environments* on page 105

Performing the Installation

Go to the Installation Guide for Runtime and locate the instructions for your chosen installation scenario. Use the information in the installation worksheet you filled out for your installation scenario to fill in the installer panels.

Prerequisites

Complete an installation worksheet for the SAP Mobile Platform installation scenario you have chosen. You will use the information from this worksheet to make selections and enter configuration information when you run the SAP Mobile Platform installer.

Task

1. The *Installation Guide for Runtime* has separate chapters for installing SAP Mobile Platform according to each of the four installation scenarios:

- *Installing SAP Mobile Platform on a Single Server*
- *Installing SAP Mobile Platform in a Simple Load-Balancing Cluster*
- *Installing SAP Mobile Platform with a Standard Microsoft Failover Cluster*
- *Installing SAP Mobile Platform with a Microsoft Failover Cluster with Shared Hosts*

2. After you complete the installation instructions for your SAP Mobile Platform installation scenario in the *Installation Guide for Runtime*, return to this document to complete any postinstallation tasks.

   The *Installation Guide for Runtime* directs you to *Completing New and Upgrade Installations* on page 40

See also

- *Completing Installation Worksheets* on page 39
- *Completing New and Upgrade Installations* on page 40
- *Adding Relay Servers or Reverse Proxies* on page 49
- *Agentry Server Host System Connectivity* on page 89
- *The Agentry Server in SAP Mobile Platform Clustered Environments* on page 105

Completing New and Upgrade Installations

After completing a new or upgrade installation, perform any postinstallation tasks needed to make your SAP Mobile Platform system fully functional.

See also

- *Completing Installation Worksheets* on page 39
- *Performing the Installation* on page 39
- *Adding Relay Servers or Reverse Proxies* on page 49
- *Agentry Server Host System Connectivity* on page 89
- *The Agentry Server in SAP Mobile Platform Clustered Environments* on page 105

Upgrade: Restoring Customized Settings in SAP Control Center Configuration Files

If you modified settings in SAP Control Center configuration files in an existing SAP Mobile Platform installation, the upgrade installer overwrites them and you must manually restore the customizations in the new files.

Unless you are certain that SAP Control Center configuration files used only default settings before you upgraded, compare those files from your SAP Mobile Platform backup with the same files after the upgrade, and manually restore any customized settings.

1. From the Windows Services Control Panel, stop the SAP Control Center service.

2. In the backup you made of your SAP Mobile Platform installation before upgrading, locate the following files:
   - `SMP_HOME\SCC-XX\bin\scc.properties`
   - `SMP_HOME\SCC-XX\services\Messaging\service-config.xml`
   - `SMP_HOME\SCC-XX\services\RMI\service-config.xml`
   - `SMP_HOME\SCC-XX\services\SCC\service-config.xml`
   - `SMP_HOME\SCC-XX\services\EmbeddedWebContainer\service-config.xml`
3. Locate the corresponding files in the upgraded SAP Mobile Platform installation.
4. Copy any changed settings from the backup files into the new files, replacing any default values.
   - Do not overwrite an entire SAP Control Center configuration file with the corresponding file from the backup. You must manually and individually update specific settings in the new files.
   - If you modified the `MaxFormContentSize` value before upgrading, its location has changed. Before upgrading, the setting was:

     ```
     -Dorg.eclipse.jetty.server.Request.maxFormContentSize
     ```

     in the `SMP_HOME\SCC-XX\bin\scc.properties` file.

     After upgrading, the setting is:

     ```
     jetty.maxFormContentSize
     ```

     in the `SMP_HOME\SCC-XX\services\EmbeddedWebContainer\service-config.xml` file.

   a) Open each file from the upgraded SAP Mobile Platform installation side-by-side with the corresponding file from the SAP Mobile Platform backup.

   b) Manually copy any changed settings from the backup file into the new file.

   c) Save the updated new file.
5. From Windows Services, restart the SAP Control Center service.

Enabling the Sample Database in a Production Installation

The SAP Mobile Platform installer creates a Windows service (SAP Mobile Platform SampleDB) that enables the sample database (sampledb) if you install with a Personal or Enterprise Development license. If you have installed with an Enterprise Server (production) license and want the sample database to be accessible, you must run a script.

Development systems are typically kept separate from production systems. The sample database is provided for developers to use in a development system. The Enterprise Server license is for production systems, so the SAP Mobile Platform installer does not create the Windows service that enables the sample database when you install SAP Mobile Platform with an Enterprise Server license.

If you want the sample database to be available after you have installed with an Enterprise Server license, you can run a script to create the Windows service that enables the sample database server.

1. Verify that the sample database service does not already exist on the SAP Mobile Server installation.

   Open the Windows Services control panel and look for SAP Mobile Platform SampleDB.
2. If SAP Mobile Platform SampleDB does not exist:
   a) In the file system where the SAP Mobile Server is installed, go to the *`SMP_HOME`*`\Servers\UnwiredServer\bin` directory, where *`SMP_HOME`* is the SAP Mobile Platform installation directory, down to the `MobilePlatform` folder.
   b) Run the `sampledb.bat` script.

      To create the service to start automatically, enter:

      ```
      sampledb.bat install auto
      ```

      To create the service to be started manually, enter:

      ```
      sampledb.bat install manual
      ```
   c) Start the SAP Mobile Platform SampleDB service you just created, either by using the Windows Services control panel, or by entering, at the command prompt:

      ```
      sampledb.bat start
      ```

      For more information on the **sampledb.bat** script, see *Create or Remove the Windows Service for sampledb Server (sampledb) Utility* in *System Administration*.

Installing Introscope Enterprise Manager

Install Introscope Enterprise Manager, then configure it to work with SAP Mobile Server.

1. Go to the SAP Web site at *http://service.sap.com/instguides* (login required).
2. Select **Installation & Upgrade Guides**.
3. In the left pane, expand **SAP Components > SAP Solution Manager**.
4. Select the **Release <*version*>** of SAP Solution Manager that you have installed.
5. Expand **2a Installation Using SW Prov. Mgr**.
6. Click the **Wily Introscope Setup Guide <*version*>** for the version of Introscope that you are using.
7. Install Introscope Enterprise Manager on an SAP Mobile Server node, or any other machine.
8. (Optional) To enable the Introscope agent in the SAP Mobile Server node to connect to Introscope Enterprise Manager, in a command prompt window, navigate to *`SMP_HOME`*`\Servers\UnwiredServer\bin` and run:

   ```
   configure-introscope-agents.bat -host:<EM-host> -port:<EM-port>
   ```

where *<EM-host>* is where you have installed Introscope Enterprise Manager and *<EM-port>* is the port number for Introscope Enterprise Manager to use.

9. If Introscope agents have been disabled during SAP Mobile Server installation or upgrade, run (from `SMP_HOME\ Servers\UnwiredServer\bin`):

   ```
   configure-introscope-agents.bat -install
   ```

10. Restart SAP Mobile Server fom the desktop shortcut or Windows start menu, but not from the Windows Service panel.
11. Start Introscope Enterprise Manager and verify that agents for DotNet Process and SMP Java Server are enabled.

For background information on use of Wily Introscope with SAP products, see *SAP Note 797147* (login required).

SAP Solution Manager Overview

SAP Solution Manager provides tools that you can use to manage SAP Mobile Platform as part of your overall SAP landscape.

Use SAP Control Center to configure a connection from SAP Mobile Server to SAP Solution Manager. Once connected, you can use the Solution Manager interface to view change records, as well as perform end-to-end traces, and exception and workload analysis.

For more detailed information about configuring SAP Solution Manager to be used in conjunction with SAP Mobile Platform, see *Maintenance of SAP Mobile Platform in the System Landscape*.

For complete SAP Solution Manager documentation, see *SAP Solution Manager Setup*.

Configuring SAP Solution Manager URL

Define and maintain a URL definition associated with an SAP Solution Manager instance for each application in the landscape. This endpoint is used to upload the business transaction XML generated by the client device platforms in an end-to-end trace session.

1. In the left navigation pane, select **Configuration**.
2. In the right administration pane, click the **General** tab.
3. From the menu bar, select **Components**.
4. Select **Solution Manager** and click **Properties**.
5. In the Solution Manager Component Property dialog, enter the URL associated with the appropriate SAP Solution manager.

SAP Control Center Postinstallation Checklist

SAP Control Center is the remote SAP Mobile Platform runtime administration tool. By default, the installer configures SAP Control Center automatically for the SAP Mobile Platform environment.

Task	Complete?
To avoid security exceptions when launching SAP Control Center, set up browser security certificates. See *Setting Up Browser Certificates for SAP Control Center Connections* in *SAP Control Center for SAP Mobile Platform*.	
Log in to SAP Control Center using the default supAdmin role with the password you configured during installation. See *Logging in to SAP Control Center with an Installer-Defined Password* in *SAP Control Center for SAP Mobile Platform*.	
Confirm that all server nodes are visible in the left navigation pane. You must manually register any missing nodes, so they can be administered remotely. See *Adding or Updating SAP Mobile Server Registration Properties* in *SAP Control Center for SAP Mobile Platform*.	
Replace the PreConfiguredUserLoginModule with a new security provider for the admin security configuration on the default domain. See *Making "Admin" Security Configuration Production-Ready* in *Security*.	

Security Postinstallation Checklist

Configuring security after installing runtime components is dependent upon the successful completion of SAP Control Center postinstallation tasks. Perform the security postinstallation tasks once SAP Control Center is functionally stable.

Task	Complete?
Secure the infrastructure of data tier components. See *Securing the Data Infrastructure* in the *Security* guide.	
Prepare the runtime environment according to your backup and recovery strategy, and secure the identified backup artifacts. See *Backup and Recovery* in the *System Administration* guide.	
Secure the data tier databases by changing default passwords and encrypted data in the database file. See *Securing Data Tier Databases* in the *Security* guide.	
(Upgrade) Create and map an SUP Push User logical role for each security configuration used to authorize incoming push notifications. See *Mapping DCN or Push Roles to a User Name Defined In PreconfiguredUserLoginModule* in the *Security* guide.	

Once these are complete, perform any other necessary security administration tasks. See *Securing Data in Motion Quick Start* and *Securing Access Quick Start* in *Security*.

EIS Driver and SSO Postinstallation Checklist

Because EIS drivers are used in both preproduction and production environments, their setup is time-sensitive. Before connecting to data sources from SAP Mobile Platform, ensure that these drivers are installed and configured correctly.

Task	Completed?
Download and install all required drivers and libraries for your EIS type.	
Configure the driver in both SAP Control Center and SAP Mobile WorkSpace - Mobile Business Object Development. See *Data Source Connections* in *System Administration*, and *Creating a Data Source Connection Profile* in *SAP Mobile WorkSpace - Mobile Business Object Development*.	
If you are configuring a driver to use SSO, also install libraries, required security artifacts, and ensure the correct driver properties and values are configured. Remaining SSO tasks can be completed as part of routine security administration. See *Single Sign-on (SSO) Quick Start* in the *Security* guide.	

Preparing to Connect to JDBC Databases

To enable SAP Mobile Server connections to Oracle, DB2, and Microsoft SQL Server databases, download the appropriate JDBC driver and install it on each SAP Mobile Server host.

1. Download the JDBC driver.

JDBC driver	URL
Oracle	*http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html*
DB2	*http://www-306.ibm.com/software/data/db2/express/download.html*
SQL Server	*http://msdn.microsoft.com/en-us/data/aa937724.aspx*

2. Install the JDBC driver.
   a) Copy JDBC driver files to the `SMP_HOME\Servers\UnwiredServer\lib\3rdparty\` directory.
   b) Restart SAP Mobile Server.
   c) Repeat these steps on each node in the SAP Mobile Server cluster.

See also

- *Preparing to Connect to SAP using Java Connectors* on page 46
- *Preparing Your SAP Environment for Single Sign-on* on page 48

Preparing to Connect to SAP using Java Connectors

SAP Mobile Server can use Java Connectors (JCo) to connect to the SAP EIS. With the correct security setup, you can also implement single sign-on (SSO) authentication.

Prerequisites

You must have an SAP account to access the SAP Web site and download libraries and utilities.

Task

After installing the required SAP files, see *Single Sign-on (SSO) Quick Start* in the *Security* guide.

See also

- *Preparing to Connect to JDBC Databases* on page 45
- *Preparing Your SAP Environment for Single Sign-on* on page 48

Installing the SAPCAR Utility

Unzip and install the latest **SAPCAR** utility on your SAP Mobile Server or SAP Mobile WorkSpace host. You can use **SAPCAR** to extract the contents of compressed SAP files, for example, RFC and cryptographic library files.

The installation package is available to authorized customers on the SAP Service Marketplace. There are different distribution packages for various hardware processors. Select the package appropriate for your platform.

1. Go to the SAP Web site at *http://service.sap.com/swdc* (login required).
2. From the SAP Download Center, navigate and log in to **Support Packages and Patches** > **Browse our Download Catalog** > **Additional Components**.
3. Select **SAPCAR**.
4. Select the current version, for example, **SAPCAR 7.20**, then download the appropriate **SAPCAR** for your platform.

See also

- *Installing the SAP Cryptographic Libraries* on page 47

Installing the SAP Cryptographic Libraries

Configure Secure Network Communications (SNC) for SAP Mobile Server SAP JCo connections. SNC may be required by your SAP EIS, if you are using SSO2 tokens or X.509 certificates for connection authentication.

Prerequisites

Download and install the **SAPCAR** utility, which is required to extract the contents of the cryptographic library.

Task

Unzip and install the contents of the latest SAP Cryptographic archive on your SAP Mobile Server host. There are different distribution packages for various hardware processors.

Make sure you are installing the correct libraries for your environment, and into folders based on the architecture of your machine.

1. Go to the SAP Web site at *http://service.sap.com/swdc* (requires login) and download the latest SAP cryptographic library suitable for your platform.
   a) Navigate to **Installations and Upgrades > Browse our Download Catalog > SAP Cryptographic Software > SAPCryptolib for Installation > SAPCRYPTOLIB <version>**.
   b) Select and download the platform-specific file.
2. Create a directory into which to unzip the Cryptographic zip file. For example: `C:\sapcryptolib`.
3. Copy the appropriate Windows cryptographic library for your machine (for example, SAPCRYPTOLIB<version>.SAR) to the `C:\sapcryptolib` directory.
4. Open a command prompt and navigate to `C:\sapcryptolib`.
5. Extract the SAR file. For example:

   ```
   SAPCAR_4-20002092.EXE -xvf C:\SAPCRYPTOLIB<version>.SAR -R C:\sapcryptolib
   ```
6. Copy the following into the `C:\sapcryptolib` directory:
   - For Itanium 64-bit processors, copy the `ntia64` subdirectory contents.
   - For Intel 64-bit processors, copy the `nt-x86_64` subdirectory contents.
   - For Intel 32-bit processors, copy the `ntintel` subdirectory contents.
7. Delete the corresponding subdirectory when files have been moved.
8. If you have installed SAP Mobile SDK, you must add the SECUDIR variable to the following batch file: *SMP_HOME*`\MobileSDK<version>\Eclipse\MobileWorkSpace.bat`.

See also

- *Installing the SAPCAR Utility* on page 46

Preparing Your SAP Environment for Single Sign-on

Verify that the SAP enterprise information system (EIS) is configured correctly to accept SSO connections from SAP Mobile Server.

1. Set all parameters for the type of credentials accepted by the server:
   - SSO2 token – verify everything is set properly with the SSO2 transaction.
   - X.509 certificate – set up, import, and verify certificates using the Trust Manager (transaction STRUST).
2. Use the ICM configuration utility to enable the ICM HTTPS port.
3. Set the type of authentication to enable communication over HTTPS.
   - Server authentication only – the server expects the client to authenticate itself using basic authentication, not SSL
   - Client authentication only – the server requires the client to send authentication information using SSL certificates. The ABAP stack supports both options. Configure the server to use SSL with client authentication by setting the ICM/HTTPS/verify_client parameter:
     - 0 – do not use certificates.
     - 1 – allow certificates (default).
     - 2 – require certificates.
4. Use the Trust Manager (transaction STRUST) for each PSE (SSL server PSE and SSL client PSE) to make the server's digitally signed public-key certificates available. Use a public key-infrastructure (PKI) to get the certificates signed and into the SAP system.

   There are no SSO access restrictions for MBO data that span multiple SAP servers.

   See SAP product documentation at *http://help.sap.com/saphelp_aii710/helpdata/en/49/23501ebf5a1902e10000000a42189c/frameset.htm* for information about the SAP Trust Manager.
5. To enable secure communication, SAP Mobile Server and the SAP server that it communicates with must exchange valid CA X.509 certificates. Deploy these certificates, which are used during the SSL handshake with the SAP server, into the SAP Mobile Server keystore.
6. The user identification (distinguished name), specified in the certificate must map to a valid user ID in the AS ABAP, which is maintained by the transaction SM30 using table view (VUSREXTID).

See *Configuring the AS ABAP for Supporting SSL* at *http://help.sap.com/saphelp_aii710/helpdata/en/49/23501ebf5a1902e10000000a42189c/frameset.htm*

See also

- *Preparing to Connect to JDBC Databases* on page 45
- *Preparing to Connect to SAP using Java Connectors* on page 46

Adding Relay Servers or Reverse Proxies

Once the installation of the SAP Mobile Platform cluster is complete, install and configure either Relay Server or a reverse proxy, depending on the option you have selected during the design stage.

For Relay Server, you can use either the SAP Hosted Relay Service or installed Relay Server binaries. For third-party reverse proxy solutions, SAP currently recommends Apache Reverse Proxy.

See also

- *Completing Installation Worksheets* on page 39
- *Performing the Installation* on page 39
- *Completing New and Upgrade Installations* on page 40
- *Agentry Server Host System Connectivity* on page 89
- *The Agentry Server in SAP Mobile Platform Clustered Environments* on page 105

Using SAP Hosted Relay Service for Testing

The SAP Hosted Relay Service is an alternative to local Relay Server installation, for temporary use with development and test systems only. It is particularly suitable for a personal SAP Mobile Platform system.

Prerequisites

- All SAP Mobile Servers and data tier servers must be installed and running.
- The SAP Mobile Server cluster and its nodes must be registered in SAP Control Center.

Note: If the cluster or server name does not appear in the navigation pane, on the SCC console, you must register them with SAP Control Center.

Task

Subscribe online to the SAP Hosted Relay Service, and configure your SAP Mobile Platform system with the information you provide during subscription.

1. Register SAP Mobile Server with the relay service.
   a) Register or log in to the SAP Hosted Relay Service at *https://relayserver03.sybase.com/ias_relay_server/account/index.php*.

      Complete any mandatory fields, then click **Submit**.
   b) From the navigation bar on the left, click **Manage Account**.
   c) Click **Add New SAP Mobile Platform Farm**.

      Create one or more farms as required by your development or test environment.

- Select at least one farm type.
- Select **DCN farm** only if you are registering a scale-out node.
- Select **MBS farm** for Hybrid Web Container applications.
- Enter the farm name, which serves as the farm ID in SAP Control Center. The suffix `RBS` or `MBS` is appended to the end of the farm name, depending on the farm type you select.
- Enter the server name, which is used as the server node ID in SAP Control Center. The server name can contain only alphanumeric characters.

d) Click **Create Farm**.

e) Click **Configuration Instructions** from the confirmation message.

Keep this page open, or make a copy of these details so you have them available for further configuration tasks in SAP Control Center and on the client devices.

2. Use SAP Control Center to register the hosted relay service as a Relay Server.

a) Click the SAP Mobile Platform cluster, and open the **Relay Servers** tab, then click **New**.

b) Enter general configuration information from the configuration instructions, then click **Next**.

Host	relayserver.sybase.com
HTTP port	80
HTTPS port	443

c) Enter the server farm (SAP Mobile Server cluster) information.

Farm ID	Copy the farm name from the configuration instructions
Type	Messaging or Replication

d) Click **+**, then click the **farm ID** field.

e) Enter the node ID.

Use the server name you registered with the hosted relay service.

f) Enter the token.

Copy the token string from the configuration instructions and paste it in the field.

g) Click **+** to add the server node to the list, then **Finish**.

3. Create outbound enablers (RSOEs) to connect with the hosted relay service.

a) In the left pane, under **Servers > <*Mobile Server name*>**, select **Server Configuration**.

b) Select the **Outbound Enabler** tab.

c) Click **+ New** to create the RSOEs.

d) Select the RSOE details, then click **Finish**.

4. Start all RSOEs.

a) Select some or all RSOEs, then click **Start**.

The Status column should show `Running`.

b) Select one or more RSOEs, then click **Retrieve Log**.

Review log messages to ensure each RSOE is running correctly.

Record the connection property values shown on the Configuration Instructions page to share with developers and device users.

Developers must use those values to configure an SAP Mobile Server connection profile, and to set values in the Connection screen of a mobile application.

Installing Relay Server for Production Environments

A Relay Server supports most environments and application types, including those applications connecting as an HTTP client.

1. *Installing a Relay Server*

   Install each Relay Server instance on a Web server host, on the DMZ subnet.
2. *Configuring Relay Servers and Outbound Enablers*

   Configure Relay Servers and outbound enablers (OEs) to support load balancing in an SAP Mobile Server cluster.

Installing a Relay Server

Install each Relay Server instance on a Web server host, on the DMZ subnet.

Prerequisites

You must provision an appropriate Web server host for each Relay Server.

See also

- *Configuring Relay Servers and Outbound Enablers* on page 64

Installing Relay Server on Apache RedHat Enterprise

Install the Relay Server version 16.x executables and libraries on an Apache HTTP Server RedHat Enterprise (Linux) host.

Prerequisites

When installing the Linux operating system, choose to install Apache as the Web server. While you can manually install Apache after installing the operating system, these instructions do not apply in those cases.

Task

1. Copy the Relay Server archive file to the Linux server, and extract the archive into a Relay Server installation directory.

For example: `/usr/local/relayserver/`

2. Copy `dblgen16.res` from `/usr/local/relayserver/res` to `/usr/sbin` and to `/etc/httpd/modules`.
3. Copy `dbfhide` and `dbsupport` from `/usr/local/relayserver/bin64` to `/etc/httpd/modules`.
4. Copy `rshost` from `/usr/local/relayserver/relayserver/bin64` to `/etc/httpd/modules`.
5. Copy these files from `/usr/local/relayserver/lib64` to `/etc/httpd/modules`:
   - `libdbicu16_r.so`
   - `libdbicu16_r.so.1`
   - `libdblib16_r.so`
   - `libdblib16_r.so.1cp`
   - `libdbtasks16_r.so`
   - `libdbtasks16_r.so.1`
6. To activate the libraries, add the path to a file in `/etc/ld.so.conf.d`. For an Apache 2.2x 64-bit system, add the path `/etc/httpd/modules` to a file named `relay.conf` in `/etc/ld.so.conf.d`. Then run **ldconfig** as root to propagate the new library path to the system:

   a) Create a new `relay.conf` file in `/etc/ld.so.conf.d`.

   b) Add `/etc/httpd/modules` to the `relay.conf` file.

   c) Execute **ldconfig** from `/etc/ld.so.conf.d`.
7. Copy these files from `/usr/local/relayserver/relayserver/apache22/bin64` to `/etc/httpd/modules`:
   - mod_rs_ap_server.so
   - mod_rs_ap22_server.so.1
   - mod_rs_ap_client.so
   - mod_rs_ap22_client.so.1

   Add these additional files only for Server Monitor:
   - mod_rs_ap_monitor.so
   - mod_rs_ap22_monitor.so.1
   - mod_rs_ap_admin.so
   - mod_rs_ap22_admin.so.1
8. Copy the preconfigured `rs.config` file to `/etc/httpd/modules`:

   ```
   [options]
   verbosity = 5
   # Relay server peers
   #--------------------
   [relay_server1]
   enable          = yes
   ```

```
host            = 10.56.225.204
http_port       = 80
https_port      = 443
description     = Cluster Test

[relay_server2]
enable          = yes
host            = 10.56.225.241
http_port       = 80
https_port      = 443
description     = Cluster Test

#you can use relayserver1 and relayserver2 to create a multiple
relayserver.

#---------------
# Backend farms - RBS
#---------------
#---------------
# Backend farm - RBS
#---------------
[backend_farm]
enable          = yes
id              = shuaqa-cluster6.SUPFarm
#client_security = on
#backend_security= on
description     =
#-----------------
# Backend servers - RBS
#-----------------
[backend_server]
enable   = yes
farm     = shuaqa-cluster6.SUPFarm
id       = shuaqa-ms1
# uncomment mac if you want Relay Server to do MAC verification.
# the value should have only the 1 MAC address RSOE sends to relay
server.
# Look in the RSOE.log file to see what that address is
#mac      = 00-50-56-9c-00-0d
token      = RBS

[backend_server]
enable   = yes
farm     = shuaqa-cluster6.SUPFarm
id       = shuaqa-ms2
# uncomment mac if you want Relay Server to do MAC verification.
# the value should have only the 1 MAC address RSOE sends to relay
server.
# Look in the RSOE.log file to see what that address is
#mac      = 00-50-56-9c-00-0d
token      = RBS

#---------------
# Backend farm - MBS
#---------------
[backend_farm]
```

```
enable          = yes
id              = shuaqa-cluster6IMO.SUPFarm
#client_security = on
#backend_security= on
description     =
#-----------------
# Backend servers - MBS
#-----------------
[backend_server]
enable   = yes
farm     = shuaqa-cluster6IMO.SUPFarm
id       = shuaqa-ms1
# uncomment mac if you want Relay Server to do MAC verification.
# the value should have only the 1 MAC address RSOE sends to relay
server.
# Look in the RSOE.log file to see what that address is
#mac      = 00-50-56-9c-00-0d
token      = MBS

[backend_server]
enable   = yes
farm     = shuaqa-cluster6IMO.SUPFarm
id       = shuaqa-ms2
# uncomment mac if you want Relay Server to do MAC verification.
# the value should have only the 1 MAC address RSOE sends to relay
server.
# Look in the RSOE.log file to see what that address is
#mac      = 00-50-56-9c-00-0d
token      = MBS
```

Configuring and Testing Apache for Relay Server on RedHat Enterprise

Modify the Apache HTTP Server configuration and environment, as needed for Relay Server on RedHat Enterprise (Linux) host.

Prerequisites

Create a Relay Server configuration file and deploy it on the Relay Server (Apache) host.

Task

1. Modify the `httpd.conf` file to configure Apache for Relay Server usage by adding these entries:

```
LoadModule iarelayserver_client_module modules/
mod_rs_ap_client.so
LoadModule iarelayserver_server_module modules/
mod_rs_ap_server.so
<LocationMatch /cli/iarelayserver/* >
SetHandler iarelayserver-client-handler
</LocationMatch>
<Location /srv/iarelayserver/* >
SetHandler iarelayserver-server-handler
RSConfigFile /etc/httpd/modules/rs.config
</Location>
ServerName <your server name>
```

The Relay Server is now ready to test.

2. Start Apache and RelayServer from `/etc/httpd/modules` with these commands:
   - **Apache – /sbin/service httpd start**
   - **Relay Server – ./rshost –o /usr/relay.log –f rs.config**

   By default the Apache Server does not automatically start after a reboot. You must enable the service in the service configuration panel.
3. Set the SQLANY16 environment variable:
   a) Select `/etc/profile.d`.
   b) Create a new file named `sqlany16.sh`.
   c) Add the export for the SQLANY16 path: `export SQLANY16=/usr/local/relayserver`
4. Reboot the Apache server.
5. Set up the State Manager process to run as a service on the Relay Server host:
   a) Create a new folder named `bin64s` under `/usr/local/relayserver`.
   b) Copy `rshost` from `/usr/local/relayserver/bin64` to `/usr/local/relayserver/bin64s`.
   c) Open a console and select `/usr/local/relayserver/bin64`.
   d) As the current "apache_user" with appropriate permissions (be sure **rshost** and **httpd** processes run under the same user), execute this command:

   ```
   ./dbsvc -y -a apache_user -t rshost -w RelayServer -q -qc
   -f /etc/httpd/modules/rs.config -os 100k -ot /tmp/rs.log
   ```
6. Enable the Apache Server as a service in the service configuration panel, and start the service.

Note: Configuring the rshost as a service creates the service with the symbolic "K" link under `/etc/rs.d/rs5.d` (for example, `K80SA_RelayServer`). Verify that a symbolic "S" link is available for the relay server service, and if not, create it using one of these methods:

- **a.** Verify K80SA_RelayServer exists in `/etc/rc.d/rc5.d`.
  b. Verify a corresponding "S" link for the RelayServer exists.
  c. If not, execute the command: **ln -s /etc/init.d/SA_RelayServer S80RelayServer**
- If no link at all exists for RelayServer, execute **/sbin/chkconfig SA_Relayserver on**, which should create the service similar to the **dbsvc** command. This method should find and use the native **chkconfig** tool to install the service, which creates the links.

- **Logging –** If the Relay Server is running as a service the log is always located at `/tmp`. The Relay Server log file is named `ias_relay_server_host.log` by default, unless another file name was defined during State Manager service setup. The Apache logs (`access_log` and `error_log`) can be found at `/etc/httpd/log`.
- **Relay Server Status Request –** Request Relay Server status from a Web browser at **http://<relay server hostname>/srv/iarelayserver/**. If no Outbound Enabler has

connected the "Overall availability" is **None**. If one or more Outbound Enablers are connected, the "Overall availability" is **Partial** or **Full**.

Installing Relay Server on Apache SUSE Enterprise

Install the Relay Server version 16.x executables and libraries on an Apache HTTP Server SUSE Enterprise (Linux) host.

Prerequisites

When installing the Linux operating system, choose to install Apache as the Web server. While you can manually install Apache after installing the operating system, these instructions do not apply in those cases.

Task

1. Copy the Relay Server archive file to the Linux server, and extract the archive into a Relay Server installation directory.

   For example: `/usr/local/relayserver/`
2. Copy `dblgen16.res` from `/usr/local/relayserver/res` to `/usr/sbin` and to `/usr/lib64/apache2`.
3. Copy `dbfhide` and `dbsupport` from `/usr/local/relayserver/bin64` to `/usr/lib64/apache2`.
4. Copy `rshost` from `/usr/local/relayserver/relayserver/apache22/bin64` to `/etc/httpd/modules .`
5. Copy these files from `/usr/local/relayserver/lib64` to `/usr/lib64/apache2`:
   - `libdbicu16_r.so`
   - `libdbicu16_r.so.1`
   - `libdblib16_r.so`
   - `libdblib16_r.so.1cp`
   - `libdbtasks16_r.so`
   - `libdbtasks16_r.so.1`
6. To activate the libraries, add the path to a file in `/etc/ld.so.conf.d`. For an Apache 2.2x 64-bit system, add the path `/usr/lib64/apache2` to a file named `relay.conf` in `/etc/ld.so.conf.d`. Then run **ldconfig** as root to propagate the new library path to the system:

   a) Create a new `relay.conf` file in `/etc/ld.so.conf.d`.

   b) Add `/usr/lib64/apache2` to the `relay.conf` file.

   c) Execute **ldconfig** from `/etc/ld.so.conf.d`.
7. Copy these files from `/usr/local/relayserver/relayserver/apache22/bin64` to `/usr/lib64/apache2`:

- mod_rs_ap_server.so
- mod_rs_ap22_server.so.1
- mod_rs_ap_client.so
- mod_rs_ap22_client.so.1

Add these additional files only for Server Monitor:

- mod_rs_ap_monitor.so
- mod_rs_ap22_monitor.so.1
- mod_rs_ap_admin.so
- mod_rs_ap22_admin.so.1

8. Copy the preconfigured `rs.config` file to `/usr/lib64/apache2`:

```
[options]
verbosity = 5
# Relay server peers
#--------------------
[relay_server1]
enable          = yes
host            = 10.56.225.204
http_port       = 80
https_port      = 443
description     = Cluster Test

[relay_server2]
enable          = yes
host            = 10.56.225.241
http_port       = 80
https_port      = 443
description     = Cluster Test

#you can use relayserver1 and relayserver2 to create a multiple
relayserver.

#---------------
# Backend farms - RBS
#---------------
#---------------
# Backend farm - RBS
#---------------
[backend_farm]
enable          = yes
id              = shuaqa-cluster6.SUPFarm
#client_security = on
#backend_security= on
description     =
#-----------------
# Backend servers - RBS
#-----------------
[backend_server]
enable   = yes
farm     = shuaqa-cluster6.SUPFarm
id       = shuaqa-ms1
# uncomment mac if you want Relay Server to do MAC verification.
# the value should have only the 1 MAC address RSOE sends to relay
server.
# Look in the RSOE.log file to see what that address is
#mac      = 00-50-56-9c-00-0d
token      = RBS

[backend_server]
enable   = yes
farm     = shuaqa-cluster6.SUPFarm
id       = shuaqa-ms2
# uncomment mac if you want Relay Server to do MAC verification.
# the value should have only the 1 MAC address RSOE sends to relay
server.
# Look in the RSOE.log file to see what that address is
#mac      = 00-50-56-9c-00-0d
token      = RBS

#---------------
# Backend farm - MBS
#---------------
[backend_farm]
enable          = yes
id              = shuaqa-cluster6IMO.SUPFarm
#client_security = on
#backend_security= on
description     =
#-----------------
# Backend servers - MBS
#-----------------
[backend_server]
enable   = yes
farm     = shuaqa-cluster6IMO.SUPFarm
id       = shuaqa-ms1
# uncomment mac if you want Relay Server to do MAC verification.
# the value should have only the 1 MAC address RSOE sends to relay
server.
# Look in the RSOE.log file to see what that address is
#mac      = 00-50-56-9c-00-0d
token      = MBS

[backend_server]
enable   = yes
farm     = shuaqa-cluster6IMO.SUPFarm
id       = shuaqa-ms2
# uncomment mac if you want Relay Server to do MAC verification.
# the value should have only the 1 MAC address RSOE sends to relay
server.
# Look in the RSOE.log file to see what that address is
#mac      = 00-50-56-9c-00-0d
token      = MBS
```

Configuring and Testing Apache for Relay Server on SUSE Enterprise

Modify the Apache HTTP Server configuration and environment, as needed for Relay Server on SUSE Enterprise (Linux) host.

Prerequisites

Create a Relay Server configuration file and deploy it on the Relay Server (Apache) host.

Task

1. Modify the `default-server.conf` file to configure Apache for Relay Server usage by adding these entries:

   ```
   LoadModule iarelayserver_client_module modules/
   mod_rs_ap_client.so
   LoadModule iarelayserver_server_module modules/
   mod_rs_ap_server.so
   <LocationMatch /cli/iarelayserver/* >
   SetHandler iarelayserver-client-handler
   </LocationMatch>
   <Location /srv/iarelayserver/* >
   SetHandler iarelayserver-server-handler
   RSConfigFile /etc/httpd/modules/rs.config
   </Location>
   ServerName <your server name>
   ```

   The Relay Server is now ready to test.
2. Start Apache and RelayServer with these commands:
   - **Apache – /etc/rc.d/apache2 start**
   - **Relay Server – /usr/lib64/apache2/rshost –o /usr/relay.log –f rs.config**
3. By default the Apache Server does not automatically start after a reboot, so you must change the Services for the appropriate Runlevel:

   a) Open YaST Control Center.

   b) Open the System Services (Runlevel).

   c) Select the **apache2** service and click the runlevel 3 and 5 check boxes.

- **Logging –** If the Relay Server is running as a service the log is always located at `/tmp`. The Relay Server log file is named `ias_relay_server_host.log` by default, unless another file name was defined during State Manager service setup. The Apache logs (`access_log` and `error_log`) can be found at `/etc/httpd/log`.
- **Relay Server Status Request –** Request Relay Server status from a Web browser at **http://<relay server hostname>/srv/iarelayserver/**. If no Outbound Enabler has connected the "Overall availability" is **None**. If one or more Outbound Enablers are connected, the "Overall availability" is **Partial** or **Full**.

Interactively Installing Relay Server on IIS with Scripts

(Recommended) Use quick setup scripts to interactively install Relay Server. Quick setup can be less error-prone than manual installations.

Prerequisites

Follow the prerequisites identified in the quick setup script. You cannot install Relay Server until the script verifies that the prerequisites have been met.

Task

Output of this setup script is saved to `rs-setup.log`. The existing `rs-setup.log`, IIS metabase, and Relay Server configuration files are backed up automatically.

1. Locate the quick setup script for your version of IIS, and review the `readme.txt` file for your IIS version.
   - Launch `rs-setup.bat` for IIS 6 from `*SMP_HOME*\Servers\SQLAnywhere16\MobiLink\relayserver\IIS\QuickSetup_IIS6`. For information about this script, see *http://dcx.sybase.com/index.html#sa160/en/relayserver/ml-relayserver-s-5692444.html*.
   - Launch `rs-setup.bat` for IIS 7 from `*SMP_HOME*\Servers\SQLAnywhere16\MobiLink\relayserver\IIS\QuickSetup_IIS7`. For information about this script, see *http://dcx.sybase.com/index.html#sa160/en/relayserver/ml-relayserver-s-5692444a.html*
2. Follow the prompts to install files in the correct location and to configure IIS for Relay Server use.

   The script guides you through:
   - IIS customization
   - Backup creation
   - Installation and Relay Server startup
   - Generation and launch of a Quick Reference document
   - Generation and launch of a status page
   - Launch of a SimpleTestApp client

Manually Installing Relay Server on IIS

Perform prerequisites, then install Relay Server binaries on an IIS host. Configuration steps vary, depending on whether you are using an IIS version 6 or 7 host.

Installing Required IIS Components

You must enable certain roles and features in IIS for Relay Server to function correctly.

1. Open IIS Server Manager.
2. Verify that the IIS Role is enabled.

3. Once the IIS Role is enabled, ensure the following features are installed:

   - Web Server Service
   - Common HTTP Features
   - Static Content
   - Default Document
   - Directory Browsing
   - HTTP Errors
   - ISAPI Extensions
   - HTTP Logging
   - Request Monitor
   - Request Filtering
   - Static Content Compression
   - IIS Management Console
   - IIS Management Scripts and Tool
   - (IIS 7 only) IIS 6 Management Compatibility
   - (IIS 7 only) IIS 6 Metabase Compatibility
   - (IIS 7 only) IIS 6 WMI Compatibility
   - (IIS 7 only) IIS 6 Scripting Tools
   - (IIS 7 only) IIS 6 Management Console

   Install any missing features.
4. Restart the IIS host.

Installing Relay Server Binaries on an IIS Host

Install Relay Server executables and libraries on an IIS host.

Relay Server executables and libraries are supplied in an archive file, located on the installation media at: `modules\relayserver\`

1. Identify the appropriate archive for the IIS host architecture.

   - 32-bit system — `relayserver.zip`
   - 64-bit system — `relayserver_x64.zip`
2. Copy the Relay Server archive file to the IIS host.
3. Extract all files and folders from the archive to the `wwwroot\` directory.

   The following subdirectories are created:

   ```
   wwwroot\ias_relay_server\
   wwwroot\ias_relay_server\Client\
   wwwroot\ias_relay_server\Server\
   ```
4. Modify the system Path variable on the IIS host to include the `ias_relay_server\Server\` directory.

Configuring IIS 7 for Relay Server

Configure IIS 7 to host a Relay Server.

Note: This task configures IIS for anonymous access to Relay Server. Configure appropriate security for IIS and Relay Server, based on your business requirements.

1. Back up the `applicationHost.config` file (usually located in the `System32\inetsrv\config\` directory).
2. In a text editor, open `applicationHost.config`.
3. Create a Relay Server application pool.

   Insert the following snippet in the `<applicationPools>` collection:

   ```
   <add name="RelayServer" queueLength="65535" autoStart="true"
          managedRuntimeVersion="" managedPipelineMode="Integrated">
       <processModel identityType="LocalSystem"
   idleTimeout="00:00:00"
               maxProcesses="20" pingingEnabled="false"
              pingInterval="00:00:30" pingResponseTime="00:01:30" />
       <recycling disallowOverlappingRotation="true">
           <periodicRestart time="00:00:00">
               <schedule>
                   <clear />
               </schedule>
           </periodicRestart>
       </recycling>
       <failure rapidFailProtection="false" />
       <cpu resetInterval="00:00:00" />
   </add>
   ```
4. Add the Relay Server application to the default Web site.

   Insert the following snippet in the `<site name="Default Web Site">` element:

   ```
   <application path="/ias_relay_Server"
   applicationPool="RelayServer">
       <virtualDirectory path="/"
               physicalPath="C:\Inetpub\wwwroot\ias_relay_server" />
   </application>
   ```
5. Enable Web extensions for Relay Server.

   Insert the following snippet in the `<isapiCgiRestriction>` collection.

   ```
   <add path="C:\Inetpub\wwwroot\ias_relay_server\Client
   \rs_client.dll"
           allowed="true" />
   <add path="C:\Inetpub\wwwroot\ias_relay_server\Server
   \rs_server.dll"
           allowed="true" />
   ```
6. Add Relay Server locations to the default Web site.

Insert the following snippet in the `<configuration>` element.

```
<location path="Default Web Site/ias_relay_server/client">
    <system.webServer>
        <handlers accessPolicy="Execute, Script">
        </handlers>
    </system.webServer>
</location>

<location path="Default Web Site/ias_relay_server/server">
    <system.webServer>
        <handlers accessPolicy="Execute, Script">
        </handlers>
    </system.webServer>
</location>

<location path="Default Web Site/ias_relay_server">
    <system.webServer>
        <security>
            <authentication>
                <anonymousAuthentication userName="" />
            </authentication>
            <requestFiltering>
                <requestLimits
maxAllowedContentLength="2147483647" />
            </requestFiltering>
        </security>
    </system.webServer>
</location>
```

7. Save your changes.
8. Open a Web browser, and confirm that `http://localhost:80` loads the default page correctly.

Configuring IIS 6 for Relay Server

Use the IIS Manager Console to configure IIS 6 for Relay Server.

1. Start the IIS Manager Console.
2. Create a Relay Server application pool.

   a) Right-click **Application Pools** and create a new application pool.

   b) Right-click the newly created application pool and select **Properties** to edit its properties.

   c) Open the **Performance** tab, and deselect **Shutdown Worker Processes After Being Idle**.

   d) Open the **Recycling** tab, and deselect **Recycle Worker Processes (In Minutes)**.
3. Enable Web extensions for Relay Server.

   a) Right-click the newly created Web site, `ias_relay_server`, and select **Properties** to edit its properties.

   b) Open the **Directory** tab.
   c) Set execute permissions to **Scripts And Executables**.
   d) Click **Create under Application Settings**.
   e) Select the Relay Server application pool you created.
   f) Under **Web Service Extensions**, select **Add New Web Service Extensions**, enter Extension Name and Request Files, and select **Set Extension Status to Allowed**, to allow both `rs_server.dll` and `rs_client.dll` to be run as ISAPI.

4. Configure IIS for SAP Mobile Platform Runtime device clients to communicate with Relay Server:
   a) Navigate to `\Inetpub\AdminScripts`.
   b) Run the following console command:

      ```
      cscript adsutil.vbs set w3svc/1/uploadreadaheadsize 0
      iisreset
      ```

   If you do not perform this configuration step, you see:

   ```
   Could not connect to the Server. Session did not complete.
   ```

5. Enable anonymous access, using an appropriate user name and password for an administrative group, or using `build-user IUSR_%computername%` for directory security.

   Grant permission for the user to access the IIS metabase:

   ```
   C:\Windows\Microsoft.Net\Framework\<Version>
   \aspnet_regiis.exe -ga IUSR_%computername%
   ```

Configuring Relay Servers and Outbound Enablers

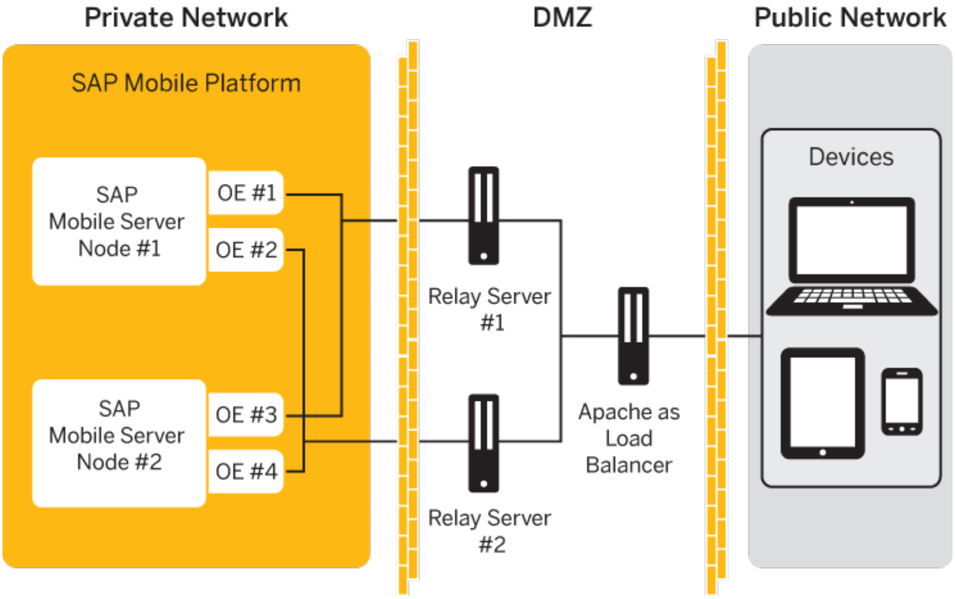
Configure Relay Servers and outbound enablers (OEs) to support load balancing in an SAP Mobile Server cluster.

Prerequisites

- All SAP Mobile Servers and data tier servers must be installed and running.
- The SAP Mobile Server cluster and its nodes must be registered in SAP Control Center.

Note: If the cluster or server name do not appear in the navigation pane, on the SCC console, you must register them with SAP Control Center.

Task



The diagram above illustrates an SAP Mobile Platform Runtime cluster with two SAP Mobile Server nodes, using two Relay Servers, with Apache acting as a load balancer. This document focuses on this configuration, while providing information on what to do differently to set up different configurations, for example:

- More than two Relay Servers
- Hardware load balancer in place of Apache

See also

- *Installing a Relay Server* on page 51

Configuring server-name to use Relay ServerSAP Mobile Server to use Relay Server

Choose a method for configuring SAP Mobile Server to use Relay Server, then generate a Relay Server configuration file. Copy the file to the Relay Server host, and distribute the same configuration file to multiple Relay Server nodes.

This task applies only to a Relay Server installed on the LAN. It does not apply to the SAP Hosted Relay Server.

Note: If you are creating a custom Relay Server configuration, see *Creating a Custom Relay Server Configuration*.

If you are using a quick configuration, continue with *Creating a Quick Configuration*.

Creating a Quick Configuration

Create a Relay Server configuration primarily with system defaults, and create outbound enabler (OE) processes for each SAP Mobile Server.

1. In the navigation pane, click the SAP Mobile Server cluster name.
2. In the administration pane, click the **Relay Servers** tab.
3. Click **Quick Configure**.
4. Specify these property values:

   Values vary for load balanced environments. If you do not configure load balancer values, outbound enablers bypass the load balancer and high availability is compromised if a direct Relay Server connection fails.

   - **Host –** if the Relay Server farm has a load balancer in front of it, the host name or IP address of the load balancer. Otherwise, the host name or IP address of a single Relay Server.
   - **HTTP port –** if the Relay Server farm has a load balancer in front of it, the port of the load balancer. Otherwise, the Relay Server HTTP port.
   - **HTTPS port –** if the Relay Server farm has a load balancer in front of it, the port of the load balancer. Otherwise, the Relay Server HTTPS port.
   - **URL suffix –** the URL suffix used by the Outbound Enabler to connect to a Relay Server. The value you set depends on whether Relay Server is installed on IIS or Apache hosts. For IIS, use `/ias_relay_server/server/rs_server.dll` . For Apache use`/srv/iarelayserver/`.
   - **Replication or Messaging or Web Service (for scale-out nodes) farm token –** the security token used by the Outbound Enabler to authenticate its connection with the Relay Server. Assign a token string (up to 2048 characters); one token can be shared by all farm types. The replication, messaging, and webservice farm token values can be the same.
   - **(Optional) Description –** a user-defined description of the Relay Server.
5. (Optional) Select **Advanced settings** and specify these property values:
   - **HTTP user –** user name for OE authentication on the Web server (Relay Server host).
   - **HTTP password –** password for OE authentication on the Web server.
6. (Optional) Configure connection values to required Internet proxy servers:
   - **Proxy server host –** host name of the Internet proxy server.
   - **Proxy server port –** connection port on the Internet proxy server.
   - **HTTP proxy user –** user name for OE authentication on the Internet proxy server.
   - **HTTP proxy password –** password for OE authentication on the Internet proxy server.
7. Click **OK** to generate a Relay Server configuration, and the OE processes for each SAP Mobile Server.

Properties in the `[backend_farm]` and `[backend_server]` sections are populated automatically, based on the SAP Mobile Server cluster name and host name.

An outbound enabler instance is automatically created for the RBS and MBS synchronization ports, and for each enabled HTTP/HTTPS port for the SAP Mobile Server host. For scale-out nodes, an outbound enabler instance is automatically created for each enabled HTTP/HTTPS port. The outbound enablers are not started.

Next

Review the values in the Relay Server configuration file, and edit if necessary.

Continue with *Generating and Modifying the Relay Server Configuration File*.

Creating a Custom Relay Server Configuration

Create a Relay Server configuration by specifying all configuration property values.

1. *Launching the Relay Server Configuration Wizard*

   Launch the Relay Server Configuration wizard to create a configuration file with customized property values.
2. *Setting Relay Server General Properties*

   Set basic connection properties for the Relay Server.
3. *Defining Server Farms and Cluster Nodes*

   Set connection properties for the SAP Mobile Server cluster and its constituent nodes.

Launching the Relay Server Configuration Wizard

Launch the Relay Server Configuration wizard to create a configuration file with customized property values.

1. In the navigation pane, click the SAP Mobile Server cluster name.
2. In the administration pane, click the **Relay Servers** tab.
3. Click **New**.

Setting Relay Server General Properties

Set basic connection properties for the Relay Server.

Prerequisites

Launch the Relay Server configuration wizard.

Task

1. Select and specify property values for either:
   - **Standalone relay server –** outbound enablers bypass the load balancer and high availability is compromised if a direct Relay Server connection fails.

  - **Host –** the host name or IP address of a single Relay Server
  - **HTTP port –** the Relay Server HTTP port
  - **HTTPS port –** the Relay Server HTTPS port

    Note: If Relay Server uses HTTPS and certificates, clients other than those using replication-based synchronization may not be able to connect: messaging applications support only HTTP, and Hybrid Web Container applications for iOS support HTTPS, but not certificates.

  - **URL suffix –** the URL suffix used by the Outbound Enabler to connect to a Relay Server. The value you set depends on whether Relay Server is installed on IIS or Apache hosts. For IIS, use `/ias_relay_server/server/rs_server.dll` . For Apache use`/srv/iarelayserver/`.

    For IIS, the value identifies the *relative* path for `rs_client.dll`. If your IIS directory structure is different, modify this value accordingly.
  - **(Optional) Description –** a user-defined description of the Relay Server.
- **Relay server farm –** configures a farm of relay servers, including values for the load balancer that is in front of the Relay Server farm. Values vary for load balanced environments.
  - **Load balancer host –** the host name or IP address of the load balancer in front of the Relay Server farm
  - **HTTP port –** the port of the load balancer
  - **HTTPS port –** the secure port of the load balancer
  - **URL suffix –** the URL suffix used by the Outbound Enabler to connect to a Relay Server load balancer. The value you set depends on whether Relay Server load balancer is installed on IIS or Apache hosts. The default for IIS is `/ias_relay_server/server/rs_server.dll` . The default for Apache is `/srv/iarelayserver/`.

    For IIS, the value identifies the *relative* path for `rs_client.dll`. If your IIS directory structure is different, modify this value accordingly.
  - **(Optional) Description –** a user-defined description of the Load Balancer.
  - **Relay server nodes within farm –** The nodes of relay server within a relay server farm. Add or remove relay server nodes:

    To add relay server nodes, specify these property values and click **+**:
    - **Host –** the host name or IP address of a single Relay Server
    - **HTTP port –** the Relay Server HTTP port
    - **HTTPS port –** the Relay Server HTTPS port

    To remove relay server nodes, select the corresponding node, then click **X**.

2. Add or remove HTTP credentials as required:

   a) Select **Configure relay server HTTP credentials**.

b) To add new credentials, specify these property values and click **+**:

- **User name –** user name for RSOE authentication on the Web server (Relay Server host).
- **Password –** password for RSOE authentication on the Web server.

c) To remove credentials from the list, select the corresponding user name, then click **X**.

3. Click **Next**.

Defining Server Farms and Cluster Nodes

Set connection properties for the SAP Mobile Server cluster and its constituent nodes.

1. Define the SAP Mobile Server cluster.

a) Specify these property values:

- **Farm ID –** a string that identifies the SAP Mobile Server cluster for which the Relay Server manages requests. This property is case-sensitive, and must match the value in the Outbound Enabler configuration.
- **Type –** the type of request managed by the Relay Server: Replication, Messaging or Webservice protocol. When configuring Relay Server Outbound Enabler properties for a scale-out node, you can select only the Webservice farm type.
- **(Optional) Description –** user-defined description for the SAP Mobile Server cluster.

b) Click **+**.

c) Repeat steps 1 and 2 to add multiple SAP Mobile Server clusters.

d) To delete a configured SAP Mobile Server cluster, select it in the list, then click the **X** button.

2. Identify each SAP Mobile Server instance in the cluster.

a) Select an existing SAP Mobile Server cluster.

b) Specify these property values:

- **Node ID –** a string that identifies the SAP Mobile Server in the cluster. This property is case-sensitive, and it must match the value in the RSOE configuration.
- **Token –** the security token used by the Outbound Enabler to authenticate its connection with the Relay Server. Assign a token string (up to 2048 characters); one token can be shared by all farm types.

c) Click **+**.

d) Repeat steps 1 and 2 to add SAP Mobile Server cluster nodes.

e) To delete a configured SAP Mobile Server node, select it in the list and click **X**.

3. Click **Next** to review your settings, or click **Finish** to exit the wizard.

The relay server configuration screen displays the Relay Server as Type STANDALONE for a standalone Relay Server or LOAD_BALANCER for a Relay Server farm.

Note: After you exit the wizard, generate the Relay Server configuration file, and copy it to each Relay Server instance to update configuration for multiple Relay Servers.

The Relay Server is registered with SAP Control Center, and can be managed from the Relay Servers tab for the SAP Mobile Server cluster.

Generating and Modifying the Relay Server Configuration File

Generate all or part of a Relay Server configuration file. Then transfer the generated file to all Relay Server hosts.

Generating a configuration file extracts the property values stored in the cluster database during the configuration process, and writes them to a file. You may still need to edit this file.

1. In the navigation pane, click the SAP Mobile Server cluster name.
2. In the administration pane, click the **Relay Servers** tab.
3. Click **Generate**.
4. Choose **Relay server configuration file**.
5. Select the parts of the file to generate:
    - The entire Relay Server configuration
    - A server node definition
    - A farm definition
6. Click **Next**, then click **Finish**.
7. Select an output target for the file.
8. Manually edit the file if necessary, and save the changes.
   For details on other manual edits that you can perform, see the Relay Server documentation at *http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.sqlanywhere.12.0.1/relayserver/relayserver12.html*.
9. To configure a Relay Server farm, apply the same changes to the configurations of remaining farm members. The configuration among all members must be identical.

Sample Relay Server Configuration File

A sample `rs.config` file for an SAP Mobile Platform cluster with a Relay Server farm with a LoadBalancer.

This is an example of an `rs.config` file you might generate for two Relay Servers that support an SAP Mobile Platform cluster made up of two SAP Mobile Servers. After generating it, you would copy it to each Relay Server host and use it to update the Relay Server configuration.

```
#--------------------
# Relay server peers
#--------------------
[relay_server]
enable          = yes
host            = 10.172.155.150
```

```
http_port       = 80
https_port      = 443
description     = Machine #1 in RS farm

[relay_server]
enable          = yes
host            = 10.172.148.40
http_port       = 5011
https_port      = 443
description     = Machine #2 in RS farm


#---------------
# Backend farms
#---------------
[backend_farm]
enable          = yes
renew_overlapped_cookie = yes
id              = supqa-serv012.obqastress2
client_security = off
backend_security= off
description     = supqa-serv012.obqastress2

#-----------------
# Backend servers
#-----------------
#supqa-serv01
[backend_server]
enable   = yes
farm     = supqa-serv012.obqastress2
id       = supqa-serv01
token    = MBS

[backend_server]
enable   = yes
farm     = supqa-serv012.obqastress2
id       = supqa-serv011
token    = MBS

#supqa-serv02
[backend_server]
enable   = yes
farm     = supqa-serv012.obqastress2
id       = supqa-serv02
token    = MBS

[backend_server]
enable   = yes
farm     = supqa-serv012.obqastress2
id       = supqa-serv021
token    = MBS
```

Deploying a Relay Server Configuration File

After you generate a Relay Server configuration file, you must copy it to each Relay Server host, to update the Relay Server configuration. All Relay Servers in a cluster SAP Mobile

Platform installation must use identical copies of the Relay Server configuration file, `rs.config`.

Prerequisites

- All SAP Mobile Servers and data tier servers must be installed and running.
- The SAP Mobile Server cluster and at least one node must be registered in SAP Control Center.
- At least one Relay Server must be registered in SAP Control Center .
- A complete configuration file must be generated.

Task

1. Edit the generated configuration file, if necessary to refine property values in the generated Relay Server configuration file.

   For complete reference details on Relay Server configuration properties, see *http://dcx.sybase.com/index.html#sa160/en/relayserver/ml-relayserver-config-file.html*.
2. Place the new `rs.config` in the same directory as `rshost.exe`.
3. To update an existing Relay Server with the new configuration, run:

   ```
   rshost –u –f Path\rs.config
   ```

   For example, on IIS, using the default configuration file location, the command is:

   ```
   rshost –u –f C:\RelayServer\SQLAnywhere12\MobiLink\relayserver
   \IIS\Bin64\Server\rs.config
   ```

Configuring the Outbound Enablers

Set up one or more outbound enablers (OEs) for each SAP Mobile Server, connecting to the load balancer, as identified in a Relay Server configuration.

Prerequisites

- At least one Relay Server must be registered in SAP Control Center.
- A complete configuration must be defined.

Task

1. *Separating the Processing of Data Services*

   You must always separate processing for different data services (that is, either messaging, replication, or HTTPS protocols) by setting up a separate Relay Servers and outbound enablers for each protocol.
2. *Configuring RSOE General Properties*

   Set general RSOE configuration properties to define the context in which the RSOE process operates.

3. *Configuring RSOE Connection Settings*

   Set connection configuration properties for an RSOE. These properties define the RSOE connection to the Relay Server.
4. *Configuring RSOE Start Options*

   Configure start options for RSOE.
5. *Generating the Relay Server Outbound Enabler Configuration File*

   To quickly and easily replicate a common outbound enabler (RSOE) configuration to multiple hosts, generate an RSOE configuration file.

Separating the Processing of Data Services

You must always separate processing for different data services (that is, either messaging, replication, or HTTPS protocols) by setting up a separate Relay Servers and outbound enablers for each protocol.

1. Create separate Relay Server farms for applications using messaging-based synchronization (MBS), replication-based synchronization (RBS), and HTTPS protocols.

   See *Creating a Custom Relay Server Configuration* on page 67, *Generating and Modifying the Relay Server Configuration* on page 70, and *Deploying a Relay Server Configuration* on page 71.
2. Set up outbound enablers for each farm you create.

   Set the Farm type to:
   - Messaging Relay Servers to handle messaging data processing.
   - Replication for Relay Servers to handle replication synchronization.
   - Scale-out Node for Relay Servers that handle HTTPS- based data processing and messaging for Web Services clients.

Configuring RSOE General Properties

Set general RSOE configuration properties to define the context in which the RSOE process operates.

1. In the navigation pane, click **Servers > <*ServerNode*>**.
2. In the administration pane, select the **Outbound Enabler** tab, then click **New**.
3. Specify these property values:
   - **Farm type –** select the type of request managed by the Relay Server: Replication, Messaging or Webservice protocol. When configuring Relay Server Outbound Enabler properties for a scale-out node, you can select only the Webservice farm type.
   - **Mobile Server port –** select the port on which RSOE manages requests.
   - **Relay server host –** if the Relay Server farm has a load balancer in front of it, the host name or IP address of the load balancer. Otherwise, the host name or IP address of a single Relay Server.

- **Relay server port –** for Relay Server farms that use a load balancer, the port of the load balancer. Otherwise, the Relay Server HTTP or HTTPS port.
- **Mobile Server farm –** select the string that identifies the SAP Mobile Server cluster, for which the Relay Server manages requests. This property is case-sensitive, and it must match the value in the Relay Server configuration.
- **Server node ID –** select the string that identifies the SAP Mobile Server in the cluster. This property is case-sensitive, and must match the value in the Relay Server configuration.

4. Click **Next**.

Configuring RSOE Connection Settings

Set connection configuration properties for an RSOE. These properties define the RSOE connection to the Relay Server.

1. Configure SSL connection options:
   - **Server Authentication –** select this option if you want Relay Server authentication for the SSL connection.
   - **Mutual Authentication –** select this option if you want mutual authentication for the SSL connection.
   - **Key Alias –** this option is only enabled if you selected mutual authentication. Select an outbound enabled identify key alias whose public certificate is trusted by the Relay Server.
2. Specify these property values:
   - **Certificate file –** select this option and choose the `.CRT` file used to authenticate the RSOE to Relay Server. You can choose this file only if you have already loaded it into the SAP Mobile Server certificate store and your Relay Server Port selection is HTTPS:443 in General Properties.
3. Specify these property values:
   - **HTTP user –** select the user name for RSOE authentication on the Web server (Relay Server host).
   - **HTTP password –** enter the password for RSOE authentication on the Web server.
4. If RSOE connections to the Relay Server must pass through an Internet proxy server, specify these property values:
   - **Proxy server –** select the Internet proxy server.
   - **HTTP proxy user –** select the user name for RSOE authentication on the proxy server.
   - **HTTP proxy password –** type the password for RSOE authentication on the proxy server.

Configuring RSOE Start Options

Configure start options for RSOE.

1. Enable an option:
   a) Select the box that corresponds to each name.
   b) Set a value.
2. Click **OK**.
3. Ensure the process starts by viewing the Status column of the Outbound Enablers tab.

See also

- *Generating the Relay Server Outbound Enabler Configuration File* on page 76

Outbound Enabler Start Options Reference

Review available outbound enabler start options, which affect outbound enabler logging. Each outbound enabler has its own log file that you can retrieve in SAP Control Center.

Option	Default	Description
Verbosity level	0	Sets log file verbosity values: • 0 – log errors only. Use this logging level for deployment. • 1 – session-level logging. This is a higher level view of a session. • 2 – request-level logging. Provides a more detailed view of HTTP requests within a session. • 3 - 5 – detailed logging. Used primarily by Technical Support.
Reconnect delay	5	Delay before retry after connection fails.
Maximum output file size	10KB	Maximum log file size.
Truncate log file	None	Delete the log file at RSOE startup.

Option	Default	Description
Advanced	None	User-defined value for start parameters. See *Outbound Enabler* in *SQL Anywhere 12.0.1* at *http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.sqlanywhere.12.0.1/relayserver/ml-relayservers-6039420.html* .

Generating the Relay Server Outbound Enabler Configuration File

To quickly and easily replicate a common outbound enabler (RSOE) configuration to multiple hosts, generate an RSOE configuration file.

Administrators can use SAP Control Center to configure an initial RSOE for development. Once a configuration proves valid and stable, the administrator can generate the RSOE configuration file, then use `regRelayServer.bat` to apply it to SAP Mobile Server hosts.

1. In the left navigation pane of SAP Control Center, click the SAP Mobile Server cluster name.
2. In the right administration pane, click the **Relay Servers** tab.
3. Select one or more relay server configurations.
4. Click **Generate**.
5. Choose **Outbound enabler configuration XML file**, then click **Next**.
6. Click **Finish**.
7. Select an output target for the file.

See also

- *Configuring RSOE Start Options* on page 75

Configuring State Manager as a Service

Set up the State Manager process to run as a service on the Relay Server host.

This task uses the **dbsvc** utility on the Web server host, which is installed from the Relay Server archive file, supplied on the SAP Mobile Platform Runtime installation media.

1. Open a command shell window, and set the current directory to the location of the **dbsvc** executable.
2. Enter the following at the prompt.

   - IIS host (Windows)

     ```
     dbsvc -as -s auto -t rshost -w SUPRelayServer "C:\Inetpub
     \wwwroot\ias_relay_server\Server\rshost.exe" -q -qc -f
     ```

```
"C:\Inetpub\wwwroot\ias_relay_server\Server\rs.config" -
o "C:\SAP\logs\rs.log"
```

Note: The log file path you specify with **-o** must exist, before you invoke **dbsvc**.

- Apache host (Linux)

  ```
  dbsvc -y -a apache_user -t rshost -w SUPRelayServer -q -
  qc -f /apache_install/modules/rs.config -os 100K -ot /
  tmp/rs.log
  ```

Substitute parameter values shown here to match your configuration.

This command configures the State Manager process (rshost) as a service.

3. To start the State Manager service:

- From the Windows Services Control Panel, right-click **SQL Anywhere - SUPRelayServer** and select **Start**.
- From a Linux command shell:
  a. Make sure the State Manager service runs under the same user credentials as the Apache service.
  b. Change the current directory to:
    - IIS host (Windows) `C:\Inetpubs\wwwroot\ias_relay_server\Server`
    - Apache host (Linux) `/apache_install/modules/`

  c. At the command prompt, enter:

  ```
  dbsvc -u SUPRelayServer
  ```

- You can stop the State Manager service either from the Windows Services control panel, or by entering the following at a command shell prompt:

  ```
  dbsvc -x SUPRelayServer
  ```

- You can uninstall the State Manager service by entering the following at a command shell prompt:

  ```
  dbsvc -d SUPRelayServer
  ```

Installing a Load Balancer

(Applies only to deployments with two or more Relay Servers.) You can use either a software or a hardware load balancer to get the best performance out of your Relay Servers.

You should be able to use any hardware or software load balancer with the Relay Servers you have installed to support your SAP Mobile Platform cluster. This document describes how to use Apache 2.2, with proxy load balancing.

If you are using a hardware load balancer, see the product documentation for instructions on setting it up.

If you are using a software load balancer other than Apache 2.2, see the program documentation for instructions on setting it up.

Using Relay Server with a Third-Party Load Balancer

You can use Relay Server with hardware or software load balancers to create Relay Server farms. However, sessions must be persisted.

1. To balance loads for a Relay Server farm, use a load balancer in front of the farm you create.
2. Ensure that the load balancer is configured to use session persistence (also known as sticky sessions) for all application traffic (packets).

   The load balancer must persist with the same SAP Mobile Server and Relay Server connection properties throughout the life cycle of the communication session.

Configuring Apache 2.2 as a Load Balancer

To use Apache 2.2 and later as a software load balancer, edit the Apache configuration file to enable mod_proxy and mod_proxy_balancer and set up the load-balancing features.

Prerequisites

Review the relevant Apache reference documentation:

- `mod_proxy` – *http://httpd.apache.org/docs/2.2/mod/mod_proxy.html*
- `mod_proxy_balancer` – *http://httpd.apache.org/docs/2.2/mod/mod_proxy_balancer.html*

Task

1. Install Apache 2.2 or later on a server in your DMZ.
2. Secure the server.

   Do not enable proxying until your server is secure.
3. Enable `mod_proxy`, `mod_proxy_balancer`.

   Uncomment the following lines in the Apache Web Server configuration file (`httpd.conf`):

   ```
   # mod_proxy - core module for proxying:
   LoadModule proxy_module modules/mod_proxy.so
   # mod_proxy_balancer implements clustering and load balancing:
   LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
   ```
4. On Linux, build Apache with these modules enabled.
5. Enable additional modules as needed.

   Uncomment the lines for any additional modules for which you need the functionality. See the Apache documentation at *http://httpd.apache.org/docs/2.2/mod/*.
6. On Windows, use the template below to set up load balancing between your Relay Servers.

Add the lines below to the Apache Web Server configuration file (`httpd.conf`). Replace terms in italics with actual values in your environment:

- *Apache_port* – Apache server port number.
- *Apache_srvr_name* – Apache server name.
- *host_id* – virtual host ID.
- *host_port* – virtual host port number.
- *RS_farm* – Relay Server farm ID.
- *RS#_IP* – IP address or server name for Relay Server #.
- *RS#_node* – node name for Relay Server #.
- *RS#_port* – port number for Relay Server #.
- *webserver_doc_root* – doc root in file system used by Apache Web server.

```
<VirtualHost host_id:host_port>

    ServerName Apache_srvr_name:Apache_port
    DocumentRoot webserver_doc_root

    # Enable forward proxy
    ProxyRequests On
    ProxyVia On
    <Proxy *>
        Order deny,allow
        Allow from all
    </Proxy>

    # Enable reverse proxy
    ProxyPass / balancer://RS_farm/ stickysession=X-SUP-SESSID
    ProxyPassReverse / http://RS1_IP:RS1_port/
    ProxyPassReverse / http://RS2_IP:RS2_port
    <Proxy balancer://RS_farm>
        BalancerMember http://RS1_IP:RS1_port/ route=RS1_node
        BalancerMember http://RS2_IP:RS2_port/ route=RS2_node
        # Set counting algorithm to more evenly distribute work:
        ProxySet lbmethod=byrequests
    </Proxy>

    # Enable load balancer management
    <Location /balancer-manager>
        SetHandler balancer-manager
    </Location>

    <Directory "htdocs">
        AllowOverride AuthConfig
    </Directory>

</VirtualHost>
```

Extend the example above to any number of Relay Servers by adding them to the `ProxyPassReverse` and `BalancerMember` lists.

Note: The `stickysession` parameter is required to support client request routing, also referred to as back-end server affinity, so that the load balancer always sends a device's request to the same server.

7. On Linux, use the template below to set up load balancing between your Relay Servers.

   Add the lines below to the Apache Web Server configuration file (`httpd.conf`). Replace terms in italics with actual values in your environment:

   - *RS_farm* – Relay Server farm ID.
   - *RS#_IP* – IP address or server name for Relay Server #.
   - *RS#_node* – node name for Relay Server #.
   - *RS#_port* – port number for Relay Server #.
   - *RS1#_srvr* – server name for Relay Server #.

```
# Enable forward proxy
ProxyRequests On
ProxyVia On
<Proxy *>
    Order deny,allow
    Allow from all
</Proxy>

<Proxy balancer://mycluster>
    BalancerMember http://RS1_IP:RS1_port/ route=RS1_node
loadfactor=5 route=RS1_srvr
    BalancerMember http://RS2_IP:RS2_port/ route=RS2_node
loadfactor=5 route=RS2_srvr
    # Set counting algorithm to more evenly distribute work:
    ProxySet lbmethod=byrequests
</Proxy>

# Enable reverse proxy
ProxyPass / balancer://RS_farm/ stickysession=X-SUP-SESSID
ProxyPassReverse / http://RS1_IP:RS1_port
ProxyPassReverse / http://RS2_IP:RS2_port

# Enable load balancer management
<Location /balancer-manager>
    SetHandler balancer-manager
    Order Deny,Allow
    Allow from all
</Location>
```

Configuring Apache as a Load Balancer for the EIS Back End

When you use Apache as a load balancer for the EIS back end, some of the configuration file settings are different from those for Apache as a load balancer on the front end.

1. If the load balancer connects to SAP Data Orchestration Engine – DOE to load balancer to SAP Mobile Platform cluster to device – use the settings below.

Add the lines below to the Apache Web Server configuration file (`httpd.conf`). Replace terms in italics with actual values in your environment:

- *DOE_cluster* – DOE cluster ID
- *US#_IP* – IP address for SAP Mobile Server #
- *US#_node* – node name for SAP Mobile Server #
- *US#_port* – port number for SAP Mobile Server #

```
  ProxyPass / balancer://DOE_cluster/ stickysession=X-SUP-SESSID
  ProxyPassReverse / http://US1_srvr:US1_port/doe/publish
  ProxyPassReverse / http://US2_srvr:US2_port/doe/publish
  <Proxy balancer://DOE_cluster>
   BalancerMember http://US1_srvr:US1_port/doe/publish
route=US1_node
   BalancerMember http://US2_srvr:US2_port/doe/publish
route=US2_node
```

Extend the example above to any number of SAP Mobile Servers by adding them to the `ProxyPassReverse` and `BalancerMember` lists.

2. If the load balancer connects to SAP MobileGateway – MobileGateway to load balancer to SAP Mobile Platform cluster to device – use the settings below in place of the corresponding settings in *Configuring Apache 2.2 as a Load Balancer* on page 78.

Add the lines below to the Apache Web Server configuration file (`httpd.conf`). Replace terms in italics with actual values in your environment:

- *DOE_cluster* – DOE cluster ID.
- *US#_IP* – IP address for SAP Mobile Server #.
- *US#_node* – node name for SAP Mobile Server #.
- *US#_port* – port number for SAP Mobile Server #.

Note: The only difference in the settings below, relative to the settings for DOE, is the omission of "/doe/publish" in the references to SAP Mobile Server instances.

```
  ProxyPass / balancer://DOE_cluster/ stickysession=X-SUP-SESSID
  ProxyPassReverse / http://US1_srvr:US1_port
  ProxyPassReverse / http://US2_srvr:US2_port
  <Proxy balancer://DOE_cluster>
   BalancerMember http://US1_srvr:US1_port route=US1_node
   BalancerMember http://US2_srvr:US2_port route=US2_node
```

Extend the example above to any number of SAP Mobile Servers by adding them to the `ProxyPassReverse` and `BalancerMember` lists.

Tips for Tuning Your Relay Server Configuration on IIS

Increase performance of Relay Servers hosted by IIS Web server.

Tuning your Relay Server configuration is a complex, iterative process that is beyond the scope of this document to cover in detail. This topic provides some basic tips to get you started on IIS. For instructions on implementing these tips, see the Microsoft documentation for your version of IIS.

1. Use a separate application pool for ias_relay_server\client and ias_relay_server\server.
2. For the ias_relay_server\client application pool:
   a) Disable the request queue limit.
   b) Increase the Web garden size to 4 times the number of CPU cores.
3. For the ias_relay_server\server application pool, set the Web garden size to 1.
4. Use a 64-bit Windows OS to get past the 32-bit limits nonpaged pool memory and virtual address space.
5. Monitor network performance and upgrade the network or switches if they are identified as bottlenecks.
6. Add Relay Servers to the farm.

Next

For an overview of performance tuning for Relay Server clusters, see *Performance Tuning Considerations on IIS* on page 82.

For a comprehensive guide to IIS tuning, refer to the *Internet Information Services (IIS) 7.0 Resource Kit*, written by Mike Volodarsky, Olga Londer, Brett Hill, Bernard Cheah, Steve Schofield, Carlos Aguilar Mares, and Kurt Meyer with the Microsoft IIS Team.

Performance Tuning Considerations on IIS

Many characteristics of the network environment in which SAP Mobile Platform and Relay Servers are installed can reduce performance. Understanding what these are and which are most likely to cause problems provides some guidance in how to approach performance tuning.

Traffic Factors Affecting Performance

All of the following traffic factors can affect Relay Server cluster performance:

- Persistency
- Concurrency
- Timeout
- HTTP request/response size
- Back-end farm size
- Back-end server compute time
- Relationship between transfer rates of client-Relay Server network and outbound enabler-Relay Server network

Iterative Tuning

The key objective in performance tuning is to identify the bottlenecks, or limiting points, in the overall system so you can increase capacity where needed.

In practice, you must iteratively test with a targeted set of loads for your business and factor in traffic patterns at different times of day, especially at peak levels. Identify the bottlenecks in the overall system and then tune to relieve one or more identified bottlenecks. This process shifts the bottleneck from one component to another in the overall system, while gaining overall performance on each tuning iteration.

Where to Start

If the network surrounding the Relay Servers is not saturated, your first iteration identifies the Relay Servers as the bottleneck. Determine whether the existing Relay Servers can be tuned to perform more efficiently before considering adding Relay Servers to the farm. Relay Servers might be limited in speed by the resources below.

Potential Speed Limiting Resource	Likelihood
Network I/O between Relay Servers and outbound enablers	Likely
Network I/O between Relay Servers and clients	Likely
Application pool process (IIS) and thread availability	Unlikely
CPU (including overhead on context switching and interrupts)	Very unlikely

Relay Servers can be limited in data volume by these resources.

Potential Data Volume Limiting Resource	Likelihood
Application pool process (IIS) and thread availability	Likely
Preallocated shared memory of a fixed size	Likely under heavy load
Virtual memory	Unlikely
Nonpaged pool memory consumed by HTTPS system driver and kernel	Unlikely

You can monitor all these resources via Windows performance monitor except shared memory consumption; you can view this in a shared memory report in the Relay Server log whenever you archive the log. Tuning these elements, except shared memory size, falls under general IIS tuning practices. Separating `rs_client.dll` and `rs_server.dll` into different application pools enables resource partitioning and makes fine-tuning possible.

Relay Server Pass-Through Mode for HTTP Clients

HTTP clients can connect through a Relay Server to SAP Mobile Server to use the REST services.

Common Requirements of Reverse Proxies

Reverse proxies that are used with SAP Mobile Platform must comply with specific requirements for content encoding, HTTP headers, timeouts, and the URL that is passed to SAP Mobile Platform.

A reverse proxy that is used with SAP Mobile Platform must be a straight passthrough proxy server. Ensure that any reverse proxy used:

- Does not change the content encoding of the requests or responses. Chunked transfer encoding is the required data transfer mechanism. Content-length encoding is not supported.
- Does not remove any HTTP headers.
- Sets a timeout period, if used, that is greater than the timeout used by the clients.
- Passes the resulting URL to SAP Mobile Platform Runtime in the form `http://` *`Host_Name`*`:`*`Port_No`*.

Using a Reverse Proxy in Front of Relay Server

Configure a dedicated port for the Relay Server port, and configure separate ports and contexts for each Relay Server farm.

This task implements configuration policies using an Object API application as the example. The configuration details for Hybrid Apps, OData SDK, DOE applications, and REST applications are similar.

1. Configure one port to serve each Relay Server port.

   If you map an `/smp/relayserver` context of `http://reverseProxy:8080` to `http://relayServer:80`, you must add the `/smp/relayserver` context as a prefix of the URL suffix of the Relay Server.

   For example, set `/smp/relayserver/ias_relay_server/client/rs_client.dll` as `ConnectionProperties.UrlSuffix`, and set `/smp/relayserver/ias_relay_server/client/rs_client.dll/RBSFarm` as the `ConnectionProfile.StreamParams.Url_Suffix` for synchronization. Set the `FarmId` separately for `ConnectionProperties`.
2. Configure two ports, with each port serving one Relay Server farm.

   For example, if you map the root context of `http://reverseProxy:5001` to `http://relayserver:80/ias_relay_server/client/rs_client.dll/MBSFarm`, and map the root context of `http://reverseProxy:2480` to `http://relayserver:80/ias_relay_server/`

client/rs_client.dll/RBSFarm, the Object API application then can set connection properties to connect directly to SAP Mobile Server.

3. Configure two contexts, with each context serving one Relay Server farm.

   For example, if you map the /smp/message context of http://reverseProxy:8080 to http://relayserver:80/ias_relay_server/client/rs_client.dll/MBSFarm, and map the /smp/mobilink context of http://reverseProxy:8080 to http://relayserver:80/ias_relay_server/client/rs_client.dll/RBSFarm, the Object API application needs to set the URL suffix (/smp/message for registering the application and /smp/mobilink for synchronization). This is the same as connecting to SAP Mobile Platform through a Relay Server that is not installed at the default location; the only difference is that there no FarmId for a reverse proxy.

Using a Reverse Proxy with Mutual SSL Authentication

Configure the reverse proxy to connect to SAP Mobile Server using mutual SSL authentication, then set up specific certificate requirements.

1. Configure the reverse proxy to connect to the mutual SSL port of SAP Mobile Server.
2. Configure the reverse proxy to trust the SAP Mobile Server certificate.
3. Configure the reverse proxy to use an impersonator client certificate to connect to SAP Mobile Platform. The client certificate must be mapped to the "SUP Impersonator" role for all security configurations.
4. Configure the reverse proxy to require a client certificate.
5. Configure the reverse proxy with all trusted CA certificates of SAP Mobile Platform, to accept all client certificates that can be accepted by SAP Mobile Platform.
6. Configure the reverse proxy to forward the client certificate as a SSL_CLIENT_CERT HTTP header to the SAP Mobile Server, for the server to retrieve and authenticate it.

Below is a sample configuration of an Apache reverse proxy. It maps the root context of port 8082 to https://sup-serve:8002 (the default mutual SSL port).

```
Listen 8082
<VirtualHost *:8082>
ServerName proxy-server
  # activate HTTPS on the reverse proxy
    SSLEngine on
    SSLCertificateFile "C:/Apache2.2/conf/proxy-server.crt"
    SSLCertificateKeyFile "C:/Apache2.2/conf/proxy-server.key"
    SSLCertificateChainFile "C:/Apache2.2/conf/proxy-server-ca.crt"
  # activate the client certificate authentication
    SSLCACertificateFile "C:/Apache2.2/conf/trusted-client-ca.crt"
    SSLVerifyClient require
    SSLVerifyDepth 10
    SSLProxyEngine On
    SSLProxyCACertificateFile C:/Apache2.2/conf/sup-server-ca.crt
    SSLProxyMachineCertificateFile C:/Apache2.2/conf/proxy-
```

```
client.pem
  # initialize the special headers to a blank value to avoid http
header forgeries
    RequestHeader set SSL_CLIENT_CERT ""
    <Location />
        4.add SSL_CLIENT_CERT header to forward real client
certificate
        RequestHeader set SSL_CLIENT_CERT "%{SSL_CLIENT_CERT}s"
        ProxyPass https://sup-server:8002/
        ProxyPassReverse https://sup-server:8002/
    </Location>
</VirtualHost>
```

Using Apache Reverse Proxy for HTTP Clients

For organizations with HTTP clients designed to consume SAP Mobile Server services via scale-out nodes, you can optionally implement an Apache Reverse Proxy instead of a Relay Server in your production environment.

For an example of how to implement an Apache Reverse Proxy, see:

1. *Installing Apache Reverse Proxy*

   Download and install the Apache Reverse Proxy.
2. *Configuring the Reverse Proxy with httpd.conf*

   Edit the httpd.conf file to load modules required prepare the Reverse Proxy for SAP Mobile Platform use.
3. *Mapping Proxy Ports to SMP Ports*

   For reverse proxy, you must map proxy ports to SAP Mobile Platform ports.

Installing Apache Reverse Proxy

Download and install the Apache Reverse Proxy.

1. Download Apache 2.2 from a reliable source.
2. Install the proxy according to package instructions.

Configuring the Reverse Proxy with httpd.conf

Edit the `httpd.conf` file to load modules required prepare the Reverse Proxy for SAP Mobile Platform use.

For information about running a reverse proxy in Apache, see *http://www.apachetutor.org/admin/reverseproxies*. For information about SSL and proxy modules, see *http://httpd.apache.org/docs/2.2/mod/mod_ssl.html* and *http://httpd.apache.org/docs/2.2/mod/mod_proxy.html*.

1. In a text editor, open `Apache2.2\conf\httpd.conf`.
2. Uncomment these lines to load headers, and required SSL and proxy modules.

   ```
   LoadModule headers_module modules/mod_headers.so
   LoadModule ssl_module modules/mod_ssl.so
   ```

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```

The three `proxy_*` modules are required by three proxy modes: HTTP, 1-way HTTPS, and 2-way HTTPS. The `ssl_module` is required by both HTTPS proxy modes. The `headers_module` is required by 2-way HTTPS proxy mode.

3. Add these lines to enable port 8080 as an HTTP proxy.

```
 ##############################
Listen 8080
      <VirtualHost *:8080>
        ServerName proxy-server
              ErrorLog "C:/Apache2.2/logs/error.log"
              TransferLog "C:/Apache2.2/logs/access.log"
             <Location/>
              ProxyPass http://sup-server:8000/
              ProxyPassReverse http://sup-server:8000/
             </Location>
      </VirtualHost>
##############################
```

4. Add these lines to enable port 8081 as a 1-way HTTPS proxy.

```
##############################
    Listen 8081
      <VirtualHost *:8081>
         ServerName proxy-server
                ErrorLog "C:/Apache2.2/logs/error.log"
                TransferLog  "C:/Apache2.2/logs/access.log"
              # activate HTTPS on the reverse proxy
                SSLEngine on
                SSLCertificateFile  "C:/Apache2.2/conf/proxy-
server.crt"
                SSLCertificateKeyFile  "C:/Apache2.2/conf/proxy-
server.key"
               SSLCertificateChainFile  "C:/Apache2.2/conf/proxy-
server-ca.crt"
                SSLProxyEngine On
                SSLProxyCACertificateFile C:/Apache2.2/conf/sup-
server-ca.crt
         <Location />
            ProxyPass https://sup-server:8001/
             ProxyPassReverse  https://sup-server:8001/
         </Location>
      </VirtualHost>
```

5. Add these lines to enable port 8082 as a 2-way HTTPS proxy.

```
##############################
    Listen 8082
      <VirtualHost *:8082>
         ServerName proxy-server
                ErrorLog "C:/Apache2.2/logs/error.log"
                TransferLog  "C:/Apache2.2/logs/access.log"
              # activate HTTPS on the reverse proxy
                SSLEngine on
                SSLCertificateFile "C:/Apache2.2/conf/proxy-
server.crt"
                SSLCertificateKeyFile "C:/Apache2.2/conf/proxy-
server.key"
                SSLCertificateChainFile "C:/Apache2.2/conf/proxy-
server-ca.crt"         # activate the client certificate 
authentication
                SSLCACertificateFile "C:/Apache2.2/conf/trusted-
client-ca.crt"
                SSLVerifyClient require
                SSLVerifyDepth  10
                SSLProxyEngine On
                SSLProxyCACertificateFile C:/Apache2.2/conf/sup-
server-ca.crt
                SSLProxyMachineCertificateFile C:/Apache2.2/conf/
proxy-client.pem
            # initialize the special headers to a blank  value to 
avoid http header forgeries
               RequestHeader set  SSL_CLIENT_CERT ""
              <Location />
            # add  SSL_CLIENT_CERT header to forward real client 
certificate
               RequestHeader set SSL_CLIENT_CERT "%
{SSL_CLIENT_CERT}s"
               ProxyPass  https://sup-server:8002/
               ProxyPassReverse  https://sup-server:8002/
              </Location>
         </VirtualHost>
##############################
```

6. Save the file.
7. Validate the configuration by opening a browser and testing these URLs.
    - https://proxy-server:8080/debug/app1
    - https://proxy-server:8081/debug/app1
    - https://proxy-server:8082/debug/app1

See also

- *Mapping Proxy Ports to SMP Ports* on page 89

Decrypting Certificates for HTTPS Connections

The Apache 2.2 Windows version does not support encrypted certificate key files. If the key file is encrypted, you must first decrypt it.

1. To decrypt an encrypted file, from a command prompt, run:

    `openssl rsa -in encrypted.key -out decrypted.key`
2. Use the decrypted key in the `httpd.conf` file.

Mapping Proxy Ports to SMP Ports

For reverse proxy, you must map proxy ports to SAP Mobile Platform ports.

For reverse proxy information, see:

- *SAP Control Center for SAP Mobile Platform*, see:
  - *Configuring Web Container Properties*
  - *Configuring a Replication Listener*
  - *Configuring Messaging Properties*
  - *Setting Relay Server General Properties*
- *Security*, see *Port Number Reference*

See also

- *Configuring the Reverse Proxy with httpd.conf* on page 86

Agentry Server Host System Connectivity

Before installing the SAP Mobile Platform the host system must be configured to support the system connections the Agentry Server will require. Within Agentry the term system connection refers to the connectivity between the Agentry Server and the back end systems with which it will synchronize data. The specifics of the host system configuration in relation to the system connections will depend on the types of system connections in use.

The Agentry Server is capable of connecting to and synchronizing data with four different system types. A given application may contain multiple system connections of varying types. The four types of system connections supported by the Agentry Server are:

- **SQL Database**: This system connection type is used to connect the Server to either an Oracle or MS SQL Server database. The Agentry Server will synchronize data with a database system using ANSI SQL statements.
- **Java Virtual Machine**: This system connection type is used to connect the Server to a system via a Java API. The mobile application will contain Java code that calls into this API for the purposes of synchronizing data.
- **HTTP-XML**: This system connection type is used to connect the Agentry Server to an HTTP server. The Agentry Server will make CGI requests of the HTTP Server. Data returned based on these calls is expected to be in structured XML format. This return data is parsed and processed based on XPath statements within the mobile application.
- **File System**: This type of system connection is used to connect the Agentry Server to the operating system of the host for the Server. This type of connection allows the Server to execute commands on the host system for the purposes of data synchronization and file transfer, and to validate users against the host system where necessary.

Prior to installing the SAP Mobile Platform to the host system, the proper host system configuration should be performed in relation to the above-listed system connection types as they pertain to the needs of the application. Additional configuration of the Agentry Server itself will be needed after it has been installed. All of the supported system connection types provide for user authentication functionality.

See also

- *Completing Installation Worksheets* on page 39
- *Performing the Installation* on page 39
- *Completing New and Upgrade Installations* on page 40
- *Adding Relay Servers or Reverse Proxies* on page 49
- *The Agentry Server in SAP Mobile Platform Clustered Environments* on page 105

Establishing Connectivity: Oracle Net Service Names

Create a Net Service Name for use by an Agentry Server.

Prerequisites

Address the following items prior to performing this procedure:

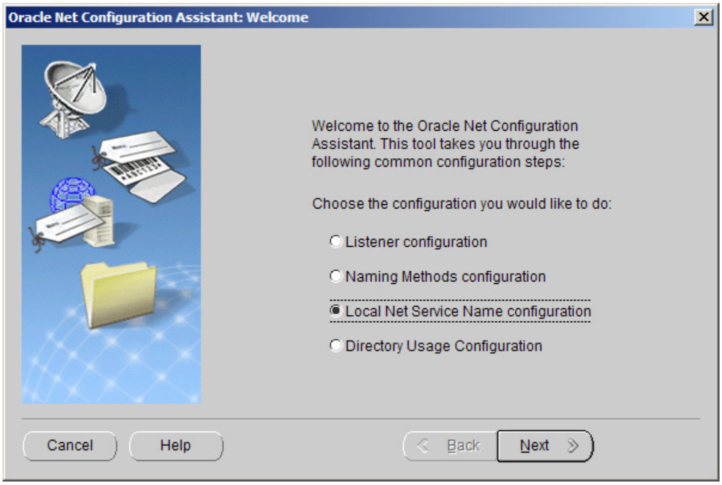
- Verify the proper version of the Oracle client software is installed to the intended host system for the Agentry Server.
- Gather the following information:
  - **Database Service Name:** The service name or global database name of the database to which the Agentry Server is to connect.
  - **Communications Protocol:** The communications protocol to use in communications between the Agentry Server and the Oracle database. May be one of TCP, TCPS, ICP, or NMP.
  - **Database Host Network Name:** The network name of the host system for the Oracle database server. This is needed only if the communications protocol used is TCP, TCPS, or NMP.
  - **Database Port Number:** The port number used to communicate with the Oracle database server. This value is only needed if the TCP or TCPS communications protocol is used.
  - **Pipe Name:** The name of the pipe for the database service. This value is only needed if the NMP communications protocol is used.
  - **Agentry Server Login and Password:** The login and password to the database that used by the Agentry Server to connect with the database server.
  - **Desired Net Service Name:** The Net Service Name by which the connection created is identified by on the Agentry Server's host system.

Task

When the Agentry Server connects with an Oracle database, the host system must have the Oracle client software installed. Using this software, an Oracle Net Service Name is created for the target Oracle database with which the Agentry Server synchronizes data. This is accomplished using the Net Configuration Assistant within the Oracle client software package.

The following procedure uses the Net Configuration Assistant as provided by version 9i of the Oracle client software. Those not familiar with this tool should review the Oracle documentation for the Net Configuration Assistant and for creating Net Service Names before proceeding.

1. Start the Oracle Net Configuration Assistant and select **Local Net Service Name configuration**.



Oracle Net Configuration Assistant: Welcome

Welcome to the Oracle Net Configuration Assistant. This tool takes you through the following common configuration steps:

Choose the configuration you would like to do:

○ Listener configuration

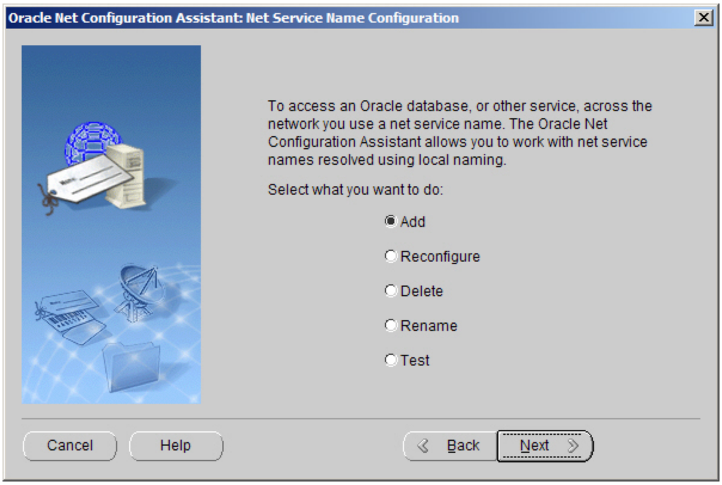
○ Naming Methods configuration

● Local Net Service Name configuration

○ Directory Usage Configuration

Click the **[Next]** button to continue.

2. Select **Add** to create the Net Service Name.



Oracle Net Configuration Assistant: Net Service Name Configuration

To access an Oracle database, or other service, across the network you use a net service name. The Oracle Net Configuration Assistant allows you to work with net service names resolved using local naming.

Select what you want to do:

● Add

○ Reconfigure

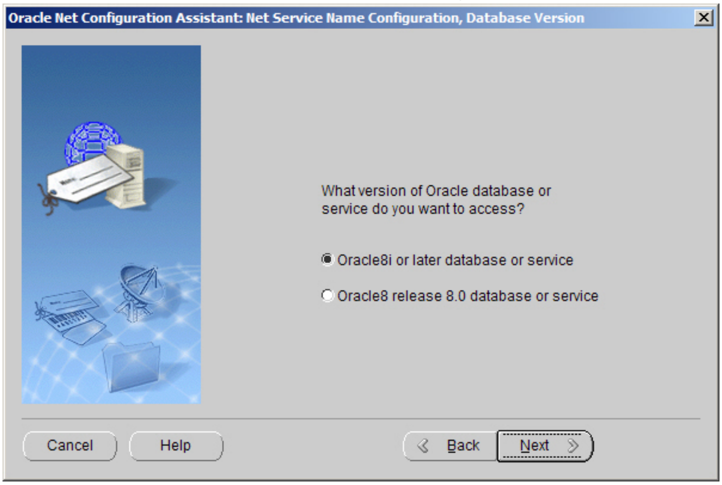
○ Delete

○ Rename

○ Test

Click the **[Next]** button to continue.

3. Select the Oracle database version used in your environment.



Oracle Net Configuration Assistant: Net Service Name Configuration, Database Version

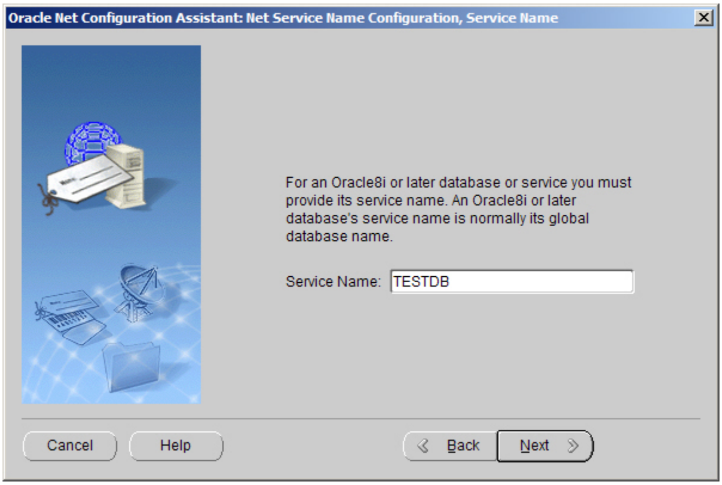
What version of Oracle database or service do you want to access?

● Oracle8i or later database or service

○ Oracle8 release 8.0 database or service

Click the **[Next]** button to continue.

4. Enter the Database Service Name to which the Agentry Server will connect.



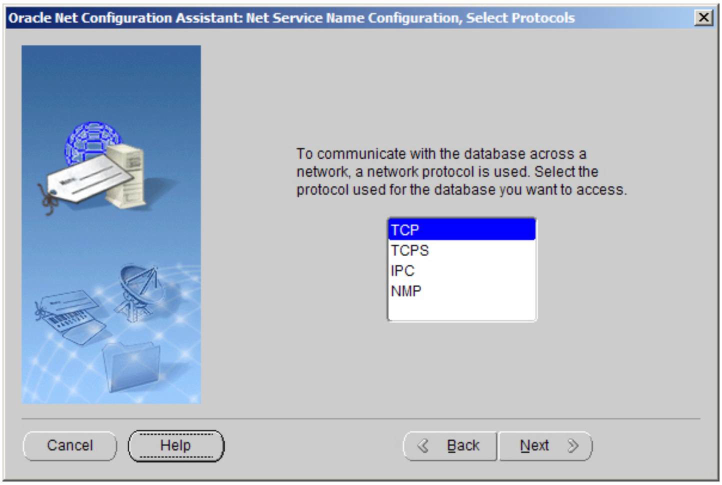
Oracle Net Configuration Assistant: Net Service Name Configuration, Service Name

For an Oracle8i or later database or service you must provide its service name. An Oracle8i or later database's service name is normally its global database name.

Service Name: TESTDB

Click the **[Next]** button to continue.

5. Select the appropriate communications protocol for your environment from those available in Oracle. Click the **[Help]** button for further information on these protocols.



Oracle Net Configuration Assistant: Net Service Name Configuration, Select Protocols

To communicate with the database across a network, a network protocol is used. Select the protocol used for the database you want to access.

TCP (selected)
TCPS
IPC
NMP

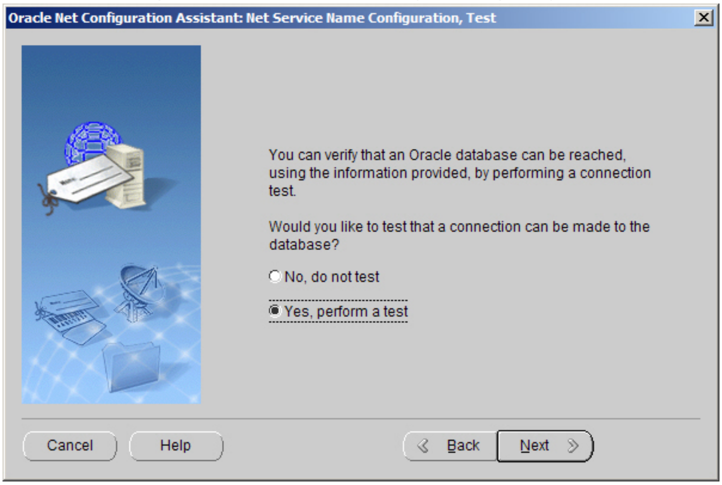
Click the **[Next]** button to continue.

6. Enter the Communications Protocol-specific configuration information.

   - **TCP or TCPS:** Enter the **Database Host Computer Name** for the database to connect to. You must also enter the **Database Port Number** by entering its value or choosing to use the default value provided.
   - **IPC:** Enter the **IPC Key** value for the local Oracle database service.
   - **NMP:** Provide the **Database Host Computer Name**. You must also enter the database pipe name or choose the default pipe name provided.

   Click the **[Next]** button to continue.

7. Test the new Net Service Name to verify all configuration options are correct and to validate the login and password used by the Agentry Server.



Oracle Net Configuration Assistant: Net Service Name Configuration, Test

You can verify that an Oracle database can be reached, using the information provided, by performing a connection test.

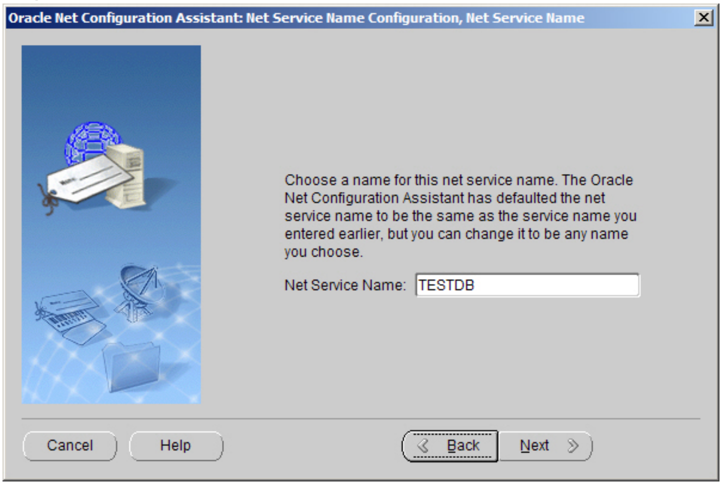
Would you like to test that a connection can be made to the database?

○ No, do not test

● Yes, perform a test

Click the **[Next]** button to continue.

8. Perform a test by following the instructions on the next screen. As an added test, change the login information to use the login and password intended for use by the Agentry Server. Once the testing is complete, click the **[Next]** button to continue.
9. Enter the **Net Service Name** through which this connection is identified on the system. Be sure to note this name as it is needed to configure the Agentry Server to connect to the target database.



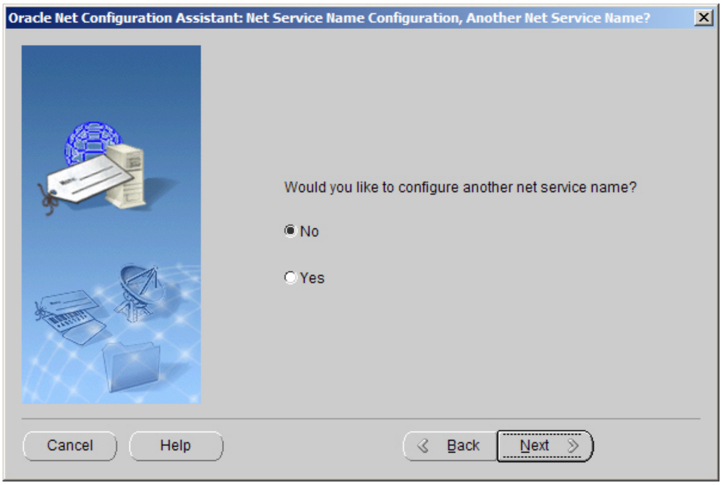
Oracle Net Configuration Assistant: Net Service Name Configuration, Net Service Name

Choose a name for this net service name. The Oracle Net Configuration Assistant has defaulted the net service name to be the same as the service name you entered earlier, but you can change it to be any name you choose.

Net Service Name: TESTDB

Click the **[Next]** button to continue.

10. Complete the creation of the new Net Service Name and close the wizard, or add another Net Service Name.



Oracle Net Configuration Assistant: Net Service Name Configuration, Another Net Service Name?

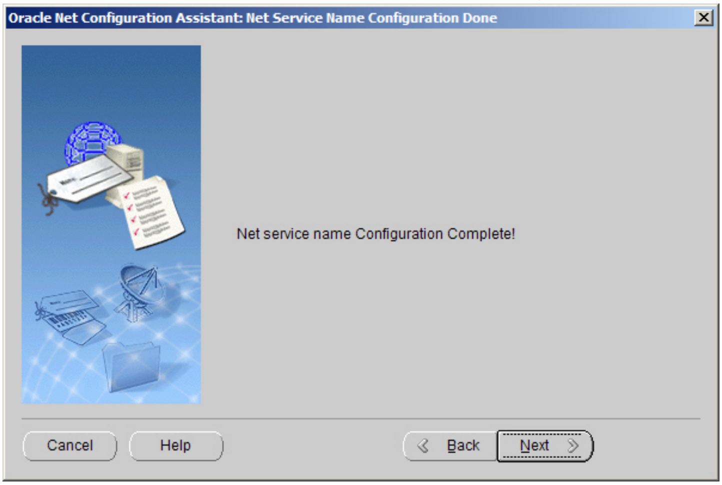
Would you like to configure another net service name?

● No

○ Yes

Selecting **Yes** will repeat this procedure. Selecting **No** will advance the wizard to the final screen, where the Net Service Name creation process is completed.

11. Complete the Net Service Name wizard by clicking the **[Next]** button on the final screen.



Oracle Net Configuration Assistant: Net Service Name Configuration Done

Net service name Configuration Complete!

At this point the main options for the Net Configuration Assistant utility are presented again. To close this utility, click the **[Finish]** button.

When this procedure is complete, an Oracle Net Service Name is created on the host system to which the Agentry Server will be installed. The Server will use this Net Service Name to connect to the database with which it will synchronize data for the mobile application.

Next

Note the Net Service Name created, as well as the login and password used by the Agentry Server to connect to the database instance. These values are needed when configuring the Agentry Server after it is installed.

Establishing Connectivity: SQL Server ODBC Connections

Create an ODBC Data Source Name (DSN), used for connections to an MS SQL Server database system.

Prerequisites

Address the following items prior to performing this procedure:

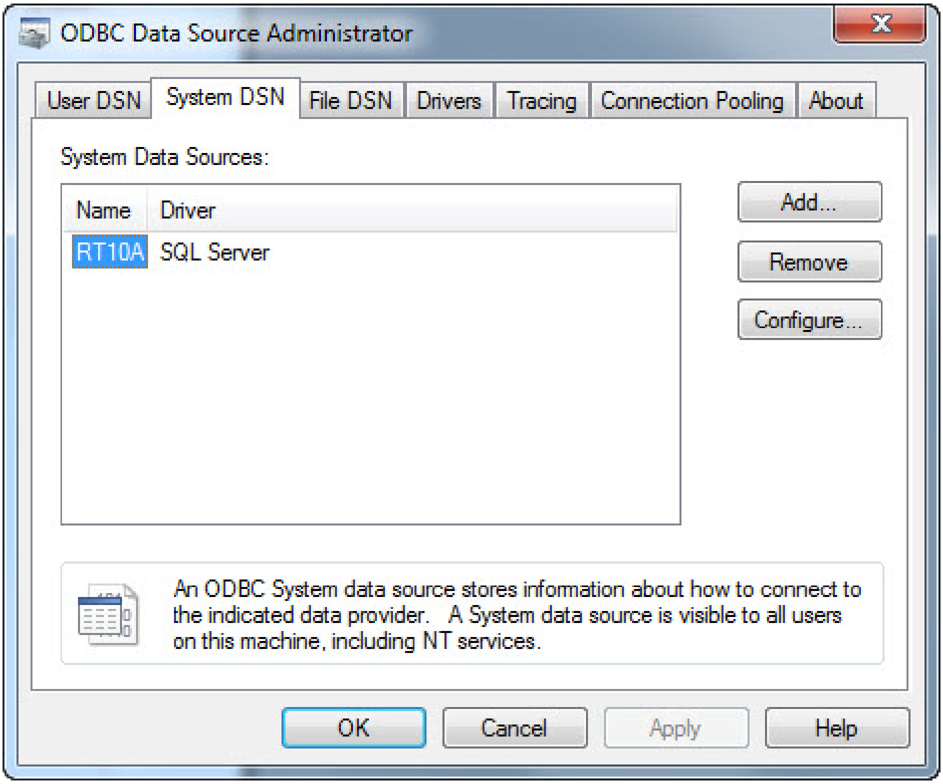
- Install or verify the presence of the proper version of the MS SQL server ODBC drivers matching the version of the SQL Server database to which the Agentry Server will connect.
- Gather the following information:
  - **DSN Name:** Determine and record the name for the DSN before the DSN is created. This is the name by which the ODBC connection for the SQL Server database is identified on the host system. It is needed when the Agentry Server is configured.

- **Server Network Name:** The network name of the host system for the SQL server database.
- **Login Authentication Method:** When creating the DSN, the option is provided to validate a database client's login and password using either Windows NT authentication, or by using SQL server authentication. Determine the proper method before creating the DSN.
- **Login and Password:** If the login authentication method will be SQL server authentication, obtain the login and password of a valid database user. If the chosen login and password are not valid, it is not possible to create the DSN.
- **Default Database:** A DSN connection is made to a SQL server database server. It is likely this server has multiple database instances. Therefore, determine the proper database instance as it will be the default instance to which a database client connects when using the DSN created in this procedure.
- **Additional Options:** There are several options available within the Add System DSN wizard that do not usually need to be modified for Agentry. However, if special circumstances in an implementation require changes to these options, make such determinations prior to creating the new DSN for the Agentry Server.

Task

A DSN is created using the Add System DSN wizard provided in the Data Sources (ODBC) utility in Windows. This information is provided for reference purposes only. Those who are not familiar with this procedure or the concepts of ODBC should review documentation provided by Microsoft before proceeding.

1. Open the Data Sources (ODBC) utility in Windows by navigating to **Start > Settings > Control Panel > Administrative Tools > Data Sources (ODBC)**. Select the **System DSN** tab.

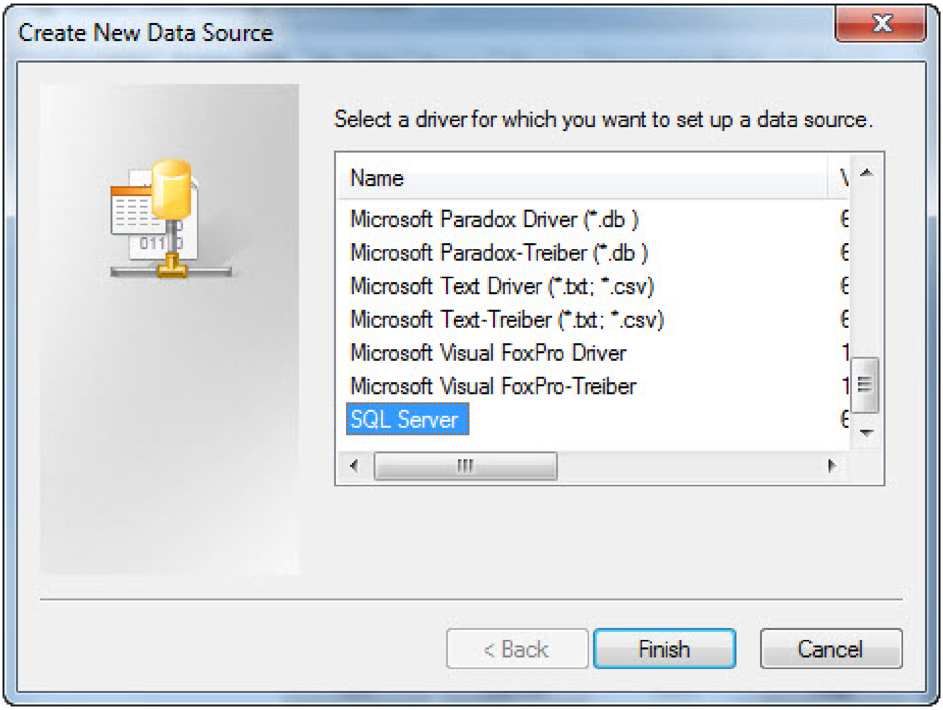


System Data Sources:

Name	Driver
RT10A	SQL Server

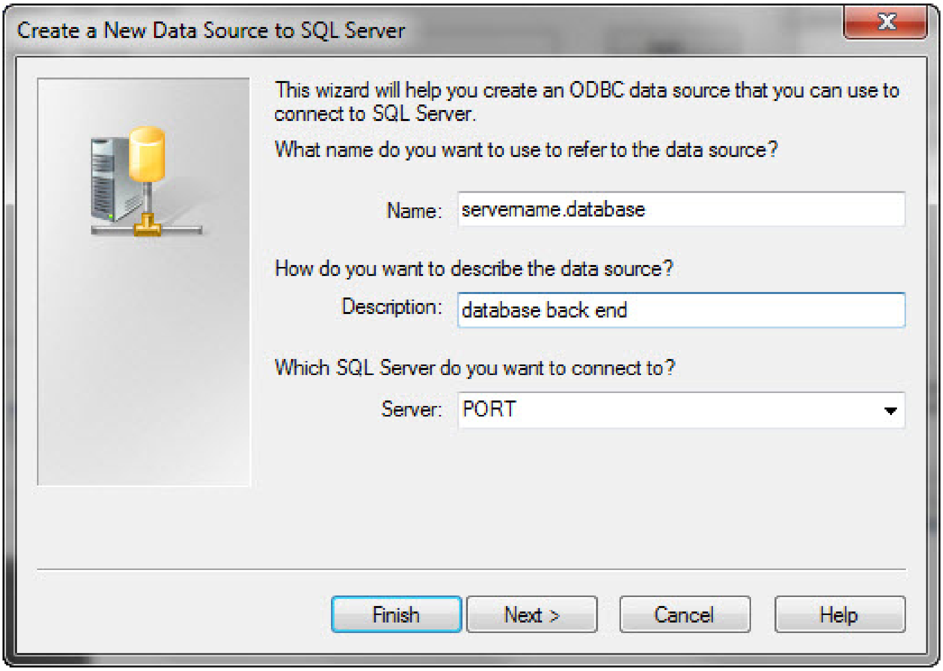
An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

2. Start the Add System DSN Wizard by clicking the **[Add]** button.


Create New Data Source
Select a driver for which you want to set up a data source.
Name
Microsoft Paradox Driver (*.db)
Microsoft Paradox-Treiber (*.db)
Microsoft Text Driver (*.txt; *.csv)
Microsoft Text-Treiber (*.txt; *.csv)
Microsoft Visual FoxPro Driver
Microsoft Visual FoxPro-Treiber
SQL Server
< Back
Finish
Cancel

Select the appropriate SQL server driver from the list and click the **[Finish]** button.

3. Enter the desired DSN Name and note this value for use during the Agentry Server installation. Also enter a Description and Network Name in the fields provided.



Create a New Data Source to SQL Server

This wizard will help you create an ODBC data source that you can use to connect to SQL Server.

What name do you want to use to refer to the data source?

Name: servername.database

How do you want to describe the data source?

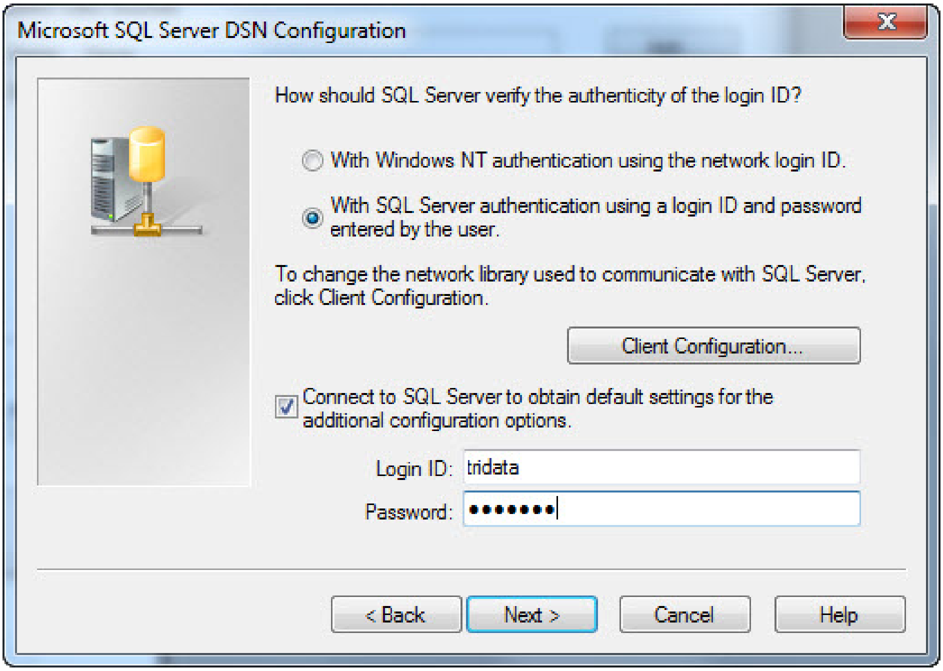
Description: database back end

Which SQL Server do you want to connect to?

Server: PORT

Enter the Server name by typing it directly or by selecting from those available on the network by opening the drop-down list for this field. Click the **[Next]** button to continue.

4. Select the authentication method for database clients using this DSN and, if SQL authentication is selected, the login and password for the SQL Server database.



Microsoft SQL Server DSN Configuration

How should SQL Server verify the authenticity of the login ID?

○ With Windows NT authentication using the network login ID.

● With SQL Server authentication using a login ID and password entered by the user.

To change the network library used to communicate with SQL Server, click Client Configuration.

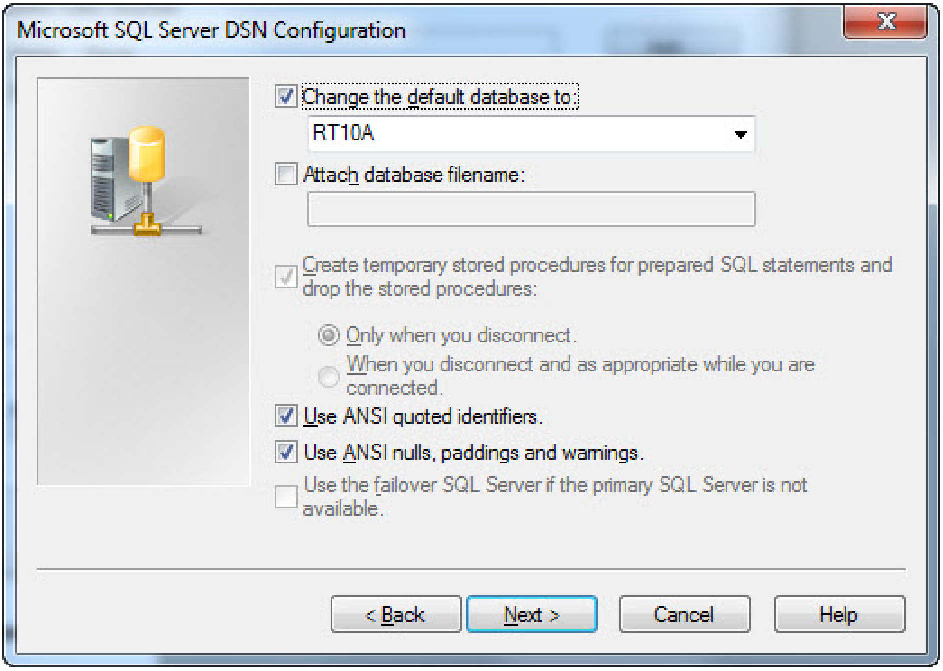
☑ Connect to SQL Server to obtain default settings for the additional configuration options.

Login ID: tridata

Password: ••••••••

Remember that the login and password are only needed if the SQL Server authentication method was selected. Click the **[Next]** button to continue.

5. Set the default database to the database with which the Agentry Server will synchronize data. In most cases the remaining options are left set to their defaults. It is recommended that these settings are changed only by an expert user that understands the purpose and resulting behavior of each setting and the overall needs of the implementation environment.


Microsoft SQL Server DSN Configuration

☑ Change the default database to:

RT10A

☐ Attach database filename:

☑ Create temporary stored procedures for prepared SQL statements and drop the stored procedures:

● Only when you disconnect.

○ When you disconnect and as appropriate while you are connected.

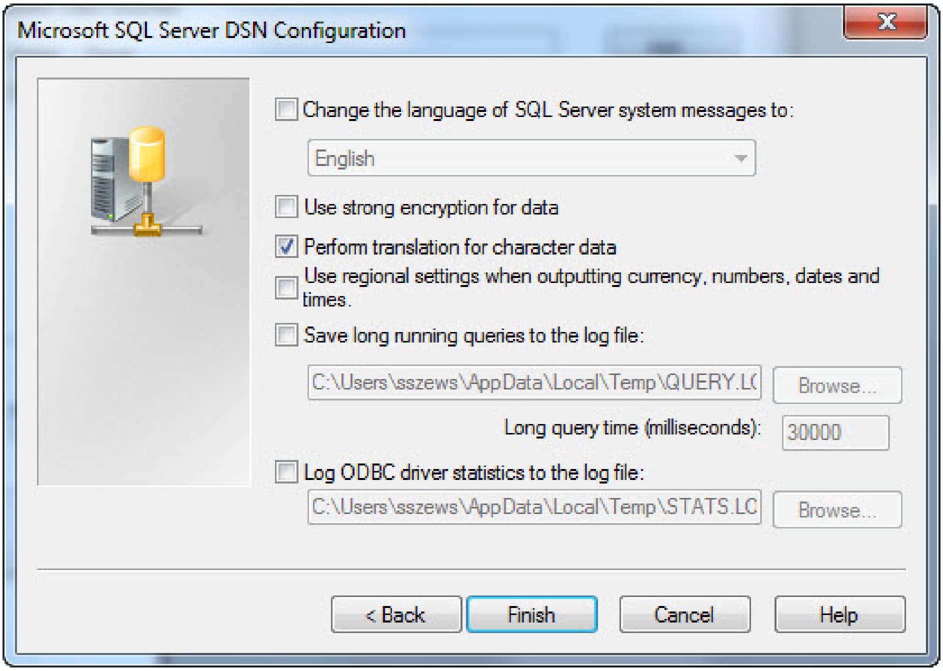
☑ Use ANSI quoted identifiers.

☑ Use ANSI nulls, paddings and warnings.

☐ Use the failover SQL Server if the primary SQL Server is not available.

Click the **[Next]** button to continue.

6. Set the final options as needed, based on the environment and administrative needs. These settings pertain primarily to locality and logging. They should only be changed by someone familiar with their purposes. The Agentry Server does not impose any requirements on the settings for these items.



Microsoft SQL Server DSN Configuration

☐ Change the language of SQL Server system messages to:

English

☐ Use strong encryption for data

☑ Perform translation for character data

☐ Use regional settings when outputting currency, numbers, dates and times.

☐ Save long running queries to the log file:

C:\Users\sszews\AppData\Local\Temp\QUERY.L(

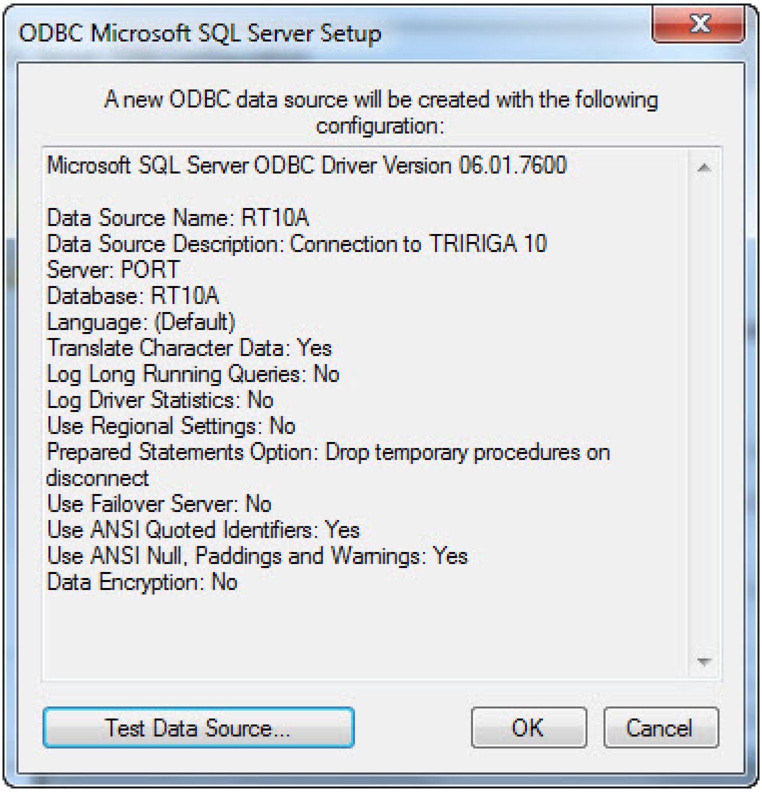
Long query time (milliseconds): 30000

☐ Log ODBC driver statistics to the log file:

C:\Users\sszews\AppData\Local\Temp\STATS.LC

Click the **[Finish]** button to complete creating the new DSN and to test it.

7. Review the summary of the DSN configuration and perform a test by clicking the **[Test Data Source]** button.



A new ODBC data source will be created with the following configuration:

Microsoft SQL Server ODBC Driver Version 06.01.7600

Data Source Name: RT10A
Data Source Description: Connection to TRIRIGA 10
Server: PORT
Database: RT10A
Language: (Default)
Translate Character Data: Yes
Log Long Running Queries: No
Log Driver Statistics: No
Use Regional Settings: No
Prepared Statements Option: Drop temporary procedures on disconnect
Use Failover Server: No
Use ANSI Quoted Identifiers: Yes
Use ANSI Null, Paddings and Warnings: Yes
Data Encryption: No

Once the test has completed successfully, click the **[OK]** button to close this wizard and return to the Data Sources (ODBC) utility. The new System DSN is listed and available to database clients needing to connect to this database, including the Agentry Server.

A new ODBC Data Source Name is created. This DSN will be used by the Agentry Server to connect with the MS SQL Server database.

Next

Make note of the Data Source Name value entered when creating the DSN. This value is needed when configuring the Agentry Server after it is installed. Also note the login and password information for SQL authentication (if selected) as this is needed by the Agentry Server.

Establishing Connectivity: Java Virtual Machine

When the Agentry Server is to synchronize data using a Java Virtual Machine system connection, the requirements of configuring the Server's intended host system depend on the needs of the back end system itself and on the nature of the interface with which the Server is to communicate.

Because of the wide array of methods of implementing Java systems, it is not possible to provide a single method for all situations. Therefore, the following tasks represent those items that will always need to be performed, as well as those that may need to be performed for a given implementation.

1. Determine the proper version of Java for the implementation. The minimum supported version of the Agentry Java API is version 1.5. The specific version will depend on how the back end system has been implemented.
2. Obtain the installer for the proper version of Java from Sun's Java web site at:

   ```
   http://java.sun.com/javase/downloads/index.jsp
   ```
3. Run the installer, which will include both the Java Development Kit (JDK) and Java Runtime Environment (JRE). Both are required components for the Agentry Server. Note the installation locations of both of these components, as they will be needed when configuring the Agentry Server after it has been installed.
4. If the API of the back end system is exposed via `.class` or `.jar` files, or other similar resources, these files should be obtained and copied to the intended host system of the Agentry Server. The location of these files should be noted, as they will be used when the Agentry Server is configured to use the JVM system connection type.
5. Obtain the host name, port number, and Java application server or interface name or identifier. Note these items for later use during the configuration of the Agentry Server.

Establishing Connectivity: HTTP-XML

When the Agentry Server is to synchronize data with an HTTP server that returns XML data, little configuration is required prior to installing the Agentry Server. The configuration required involves access to the HTTP server. The specific tasks for this vary with each implementation and/or application.

Most of the tasks involved in connecting the Agentry Server to the HTTP server occur after the Agentry Server software is installed, during the configuration phase. Review the information provided for the Agentry Server installation and configuration for details on these post-installation tasks.

Establishing Connectivity: Windows File System

The Agentry Server can establish a system connection with the file system of the host computer upon which it has been installed. This allows the Server to read and write files for the purpose of transferring those files between Clients and the file system. It also enables the Agentry Server to execute command-line processes on the host system. Finally, the Agentry Server can use this connection to read and write text files, using the contents as part of the production data for an application.

To implement a file system connection for the Agentry Server the only prerequisite of the host system is that a user account exists that the Agentry Server will run under with the proper permissions to perform the tasks required by the application. That is, this user account must have read/write access to the folders or directories on the file system with which it will be working, and the proper permissions to execute the commands the application has been developed to execute.

Once this user account is created, the Agentry Server should be executed under this account. This includes the proper configuration of the Windows Service or Linux or Unix daemon process to run the Server under this user account.

The Agentry Server in SAP Mobile Platform Clustered Environments

The Agentry Server runs in the standard SAP Mobile Platform clustered environment. When an Agentry Server is added to a node, it replicates out to all other nodes in the cluster.

The following items are important to note concerning the configuration of the Agentry Server and the network environment when running in the SAP Mobile Platform clustered environment:

- The network environment must include a load balancer that sits logically between the Agentry Clients and the Agentry Servers within the SAP Mobile Platform nodes within the cluster. The configuration of this load balancer is separate and different from the load balancer for other archetypes within SAP Mobile Platform in that Agentry client-server communications are not routed through the Relay Server.
- As a part of the communications security between the Agentry Clients and Agentry Server there is an encryption key which includes a public and private key pairing. For each Agentry Server this key pairing is unique by default. The result of this behavior is that an Agentry Client is tied to a given Agentry Server after its initial transmit. Attempting to connect with another Agentry Server with a different encryption key will result in an error and will not complete successfully. For this reason, the utility `AgentryKeyUtility` is provided with each Agentry Server installation and is used to export the encryption key from on Agentry Server instance and to then import this key into all other Agentry Server instances within a cluster.

For information on using the `AgentryKeyUtility` see the procedure *SAP Mobile Platform Clustering: Configuring Agentry Applications*. For additional guidance on configuring a load balancer, read on.

Load Balancer and SAP Mobile Platform Cluster with Agentry Applications

The load balancer configuration for Agentry applications operating within the SAP Mobile Platform cluster is a typical configuration that routes traffic to the Agentry Server, as determined by the overall configuration of the load balancer. Agentry clients contact the load balancer IP address and port, rather than an Agentry Server's IP address and port. The instructions provided for other archetypes within the SAP Mobile Platform are not applicable to Agentry applications.

Ensure that the configuration of the load balancer meets the following criteria:

- The load balancer should sit logically within the network between the Agentry Clients and the SAP Mobile Platform cluster.
- The Agentry Clients should all connect to the IP address and port of the load balancer.
- The load balancer should redirect traffic from Agentry Clients to the Agentry Servers running within the SAP Mobile Platform cluster directly. This traffic must not be routed through the Relay Server provided with SAP Mobile Platform, which is used for other archetypes within the platform, but not supported for Agentry.

There are no additional requirements beyond those of a standard load balancer implementation. The client-server communications for Agentry applications use standard TCP/IP communications as a part of the ANGEL connect type for transmit configurations.

See also

- *Completing Installation Worksheets* on page 39
- *Performing the Installation* on page 39
- *Completing New and Upgrade Installations* on page 40
- *Adding Relay Servers or Reverse Proxies* on page 49
- *Agentry Server Host System Connectivity* on page 89

SAP Mobile Platform Clustering: Configuring Agentry Applications

Configure the Agentry Server and network environment to run Agentry applications in the SAP Mobile Platform clustered environment.

Prerequisites

The following items must be addressed prior to performing this procedure:

- It is assumed that each node in the SAP Mobile Platform cluster has been installed and configured.
- Review information on the Agentry Server utility: `AgentryKeyUtility`.

Task

This procedure describes the steps necessary to migrate the Agentry Server's encryption keys for client-server communications.

1. **1.** Stop the Agentry Server application for all nodes in the cluster using the SAP Control Center.
2. **2.** In the primary node installation location of the SAP Mobile Platform, navigate to the directory: `C:\SAP\MobilePlatform\Servers\AgentryServer`

3. Execute the command `agentryKeyUtility -export=`*`fileName`* where *`fileName`* is the name of the key file to be created.
4. Copy the file created to the directory `C:\SAP\MobilePlatform\Servers\AgentryServer` of the next node in the cluster.
5. Execute the command `agentryKeyUtility -import=`*`fileName`* where *`fileName`* is the name of the key file created in the earlier step in this procedure.
6. Start the Agentry Server application for all nodes in the cluster using the SAP Control Center.
7. Deploy the application to the nodes of the cluster per documented procedures.

The encryption key has been migrated to all nodes in the cluster, allowing the Agentry Clients to connect with any of the Agentry Servers.

Next

Deploy the mobile application to the nodes in the cluster per the documented procedures. Configure the network load balancer to direct client-server communications appropriately to the Agentry Servers within the cluster. Test the application thoroughly according to standard testing procedures.

System Deployment Reference

Reference information that supports SAP Mobile Platform system deployment tasks.

Port Number Reference

Components of SAP Mobile Platform rely on communication ports for inter-process coordination, data transfer, and administrative access.

SAP Mobile Server Ports

SAP Mobile Server ports, default assignments, and protocols.

Type	Default	Protocol
Administration, SAP Mobile Server	2000 2001 (secure)	IIOP IIOPS
HTTP listeners (used for application connections, REST/OData APIs, and data change notifications)	5001 8000 8001 (secure)	HTTP HTTP HTTPS
Messaging service administration	5100	HTTP
Replication	2480 2481 (secure)	HTTP HTTPS

See also

- *SAP Mobile Platform Port Accommodation* on page 8

Data Tier Ports

Data tier server ports, default assignments, and protocols.

Type	Default	Protocol
Cache database (CDB) server, client access	5200	Command sequence on connection to SAP Mobile Server replication engine Tabular Data Stream™ (TDS) on JDBC connection TCP and UDP, when using Windows Filtering Platform (WFP)
Cluster database server, client access	5300	TDS
Monitor DB, client access	5400	TDS
domainlog DB, client access	5400	TDS

See also

- *SAP Mobile Platform Port Accommodation* on page 8

SAP Control Center Ports

Ports used by SAP Control Center services, default assignments, and protocols.

Type	Default	Protocol
RMI service	9999	TCP/IP
Messaging service	2100	TCP/IP
SCC repository database server	3638	TDS
Web container	8282 8283	HTTP HTTPS

See also

- *SAP Mobile Platform Port Accommodation* on page 8

SAP Control Center Port Assignments

Port assignments for SAP Control Center services are defined in XML configuration files.

SAP Control Center service configuration files are named `service-config.xml`, and located in subdirectories under the *SCC_HOME*`\SCC-`*X_X*`\services\` directory.

SCC Service	Configuration File Location
Messaging service	`...\services\Messaging\`
RMI service	`...\services\RMI\`
SCC repository database server	`...\services\SccSADataserver\`
Web container	`...\services\EmbeddedWebContainer\`

To change the port assigned to an SAP Control Center service, edit the `service-config.xml` file for that service.

Relay Server Ports

By default, Relay Server uses standard, IANA-assigned ports for HTTP (80) and HTTPS (443).

See also

- *SAP Mobile Platform Port Accommodation* on page 8

Reserved Ports

Ports reserved for internal use by SAP Mobile Platform components.

Type	Number	Protocol
Reserved	4343	TDS
Reserved	5011	HTTP
Reserved	6001	HTTP for SAP Introscope Agent
Reserved	8002	HTTPS

Do not use these reserved ports.

See also

- *SAP Mobile Platform Port Accommodation* on page 8

Other Ports

Significant ports that are not directly associated with an SAP Mobile Platform server component.

SySAM License Server

If you deploy SAP Mobile Platform with the served license model, all SAP Mobile Platform hosts must have network access to the license server port, on the SySAM license server host.

Type	Default	Protocol
SySAM license server	27000	

Sample Database Server

Both Personal Development Server and Enterprise Development Server Editions include a sample database, which is installed on the SAP Mobile Server host, for tutorials and simple testing.

Type	Default	Protocol
Sample database	5500	TDS

The Enterprise Server Edition includes a sample database, but it is not enabled. To enable the sample database installed with Enterprise Server Edition, see *Create or Remove the Windows Service for sampledb Server (sampledb) Utility* in *System Administration*.

See also

- *SAP Mobile Platform Port Accommodation* on page 8

Installation Directories

To ensure a successful installation, review the SAP Mobile Platform server component installation directories.

- The following tables show the high-level directories created in a single-node installation (all SAP Mobile Platform server components installed on a single host).
- In a multi-node or cluster installation, some of these directories are present only on a particular type of host.

By default, SAP Mobile Platform server components are installed in the `C:\SAP\MobilePlatform` directory. In this guide, *`SMP_HOME`* represents the SAP Mobile Platform installation directory, down to the `MobilePlatform` folder.

Table 5. SAP Mobile Platform Installation Subdirectories

Directory	Description
`_jvm`	JVM used by the uninstaller.
`sup`*`XX`*`ebflogs`	Log files created each time `installebf.bat` is run. Appears only in EBF installations upgraded from a 2.2.*x* version of SAP Mobile Platform.
`InstallLogs`	Log files created each time the SAP Mobile Platform Runtime installer is used. Use these logs to troubleshoot installer issues.
`IntroscopeAgent`	Introscope Agent for 64-bit Installations.
`JDK`*`x.x.x_x`*	JDK required by SAP Mobile Platform components. **Note:** If this is a new installation, not an upgrade from the previous version, `JDK`*`x.x.x_x`* does not exist, instead there are new folders `sapjvm_7`(32-bit platform) and `sapjvm_7-x64`(64-bit platform).
`sapjco`	SAP Java Connector files.
`scc_cert`	Certificate files for SAP Control Center.
`Servers`	SAP Mobile Platform server components.
`Servers\AgentryServer`	Agentry server.
`Servers\MessagingServer`	SAP messaging server.
`Servers\SQLAnywhere`*`xx`*	Database server for cache, cluster, and logging databases. Default database file location is the `data\` subdirectory.
`Servers\UnwiredServer`	SAP Mobile Server components.
`Servers\UnwiredServer\doe-c_clu`	SAP® Data Orchestration Engine Connector (DOE-C) Command Line Utility components. `CLU.bat` in `bin` directory starts the DOE-C console.
`Servers\UnwiredServer\doecSvlet`	SAP® Data Orchestration Engine Connector (DOE-C) runtime components.

Directory	Description
`Servers\UnwiredServer\licenses`	SySAM license files. When an unserved license is updated, copy the new files here.
`smp`*`XX`*`upgrade`	Appears only in installations upgraded from version 2.3.*x* of SAP Mobile Platform.
`ThirdParty`	License terms of third-party components included in SAP Mobile Platform.
`Uninstallers`	Uninstallers for SAP Mobile Platform Runtime components.
`Uninstallers\MobilePlatform`	SAP Mobile Platform Runtime uninstaller.
`Util`	Utilities used by the SAP Mobile Platform Runtime installer.

By default, SAP Control Center components are installed in the `C:\SAP\SCC-`*`XX`* directory.

Note: If you have other SAP products installed on the same host as SAP Mobile Server, you may have more than one version of SAP Control Center.

Table 6. SAP Control Center Installation Subdirectories

Directory	Description
`backup`	Backup files.
`bin`	Scripts to start or stop SAP Control Center management framework components.
`common`	Files shared by SAP Control Center components.
`conf`	Configuration files, including security providers for administration logins.
`ldap`	LDAP-related files.
`log`	Log files used by SAP Control Center and its console plug-ins to capture only management framework events. No SAP Mobile Platform data is captured here, except administration logins.
`plugins`	Managed resource plug-ins.
`rtlib`	Runtime library files.

Directory	Description
`sccRepoPwdChange`	SAP Control Center repository password update files.
`server`	Class and library files used by the management framework server.
`services`	Class and library files for SAP Control Center services.
`shared`	Shared class and library files.
`templates`	SAP Control Center service or plug-in template files.

Service Reference

Services are installed on each SAP Mobile Platform server host to support managing and coordinating component processes.

SAP Mobile Server Services

Services installed on an SAP Mobile Server host.

Note: Some services may not be installed on an SAP Mobile Server host, depending on the SAP Mobile Platform product option, the deployment scenario and system design, and the licensed product edition.

Service	Description
SAP Mobile Server	Top-level SAP Mobile Server process. Coordinates other processes that handle interactions with EIS services, supports messaging and synchronization service to mobile clients, and provides SAP Mobile Platform system management facilities.
SAP Control Center *X.X*	Includes processes for managing, monitoring, and controlling distributed SAP Mobile Platform server resources, and a Web app server for remote SCC console access.

Service	Description
SAP Mobile Platform SampleDB (optional)	Database server for sample database, enabled during installation only with Evaluation license, and with Personal Development Server and Enterprise Development Server licensed product editions. To enable with Enterprise Server Edition after installation, see *Create or Remove the Windows Service for sampledb Server (sampledb) Utility* in *System Administration*.

Data Tier Services

Services installed on a data tier host.

Note: Some services may not be installed on a data tier host, depending on the SAP Mobile Platform product option, the deployment scenario and system design, and the licensed product edition.

Service	Description
SAP Mobile Platform CacheDB	Database server that manages the cache database, used primarily to support mobile clients that depend on occasional synchronization of local data stores.
SAP Mobile Platform ClusterDB	Database server that manages the cluster database, which supports SAP Mobile Server runtime management and operational processes.
SAP Mobile Platform LogDataDB	Database server that manages the SAP Mobile Server logging databases (system logging and domain logging).

When the data tier is installed in a single-node system:

- The SAP Mobile Platform ClusterDB and SAP Mobile Platform LogDataDB services are not installed.
- The SAP Mobile Platform CacheDB service manages the cache database, cluster database, and logging databases.

Starting Required Services

Before beginning development, you must start required SAP Mobile Platform services.

Prerequisites

Ensure the required services are installed on the same host.

Task

By starting required services, you start the servers and dependent services. For a complete list of SAP Mobile Platform services, see *System Administration > System Reference > SAP Mobile Platform Windows Services*.

1. Click the **Start SAP Mobile Platform Services** desktop shortcut to start SAP Mobile Server and the dependent services.
2. Use the Services Control Panel to verify that the Windows service named SAP Control Center X.X is started. If it has not, start it by selecting the service and clicking **Start**.

Starting and Stopping SAP Mobile Server

You can start and stop SAP Mobile Server in different ways, depending on the use context.

Review this table to understand which method you should use.

Method	Use When	Services Started or Stopped
SAP Mobile Server list in SAP Control Center	Stopping or starting SAP Mobile Server nodes. You cannot start scale out SAP Mobile Server nodes from SAP Control Center	SAP Mobile Server service only
Desktop shortcut	Stopping or starting SAP Mobile Server locally	All runtime services installed on that host
Windows Services panel	Stopping or starting SAP Mobile Server locally	Any combination of individual services that require stopping

Note: You cannot start a Scale-out node from SAP Control Center. If you stop a Scale-out node, you must start it manually.

Index