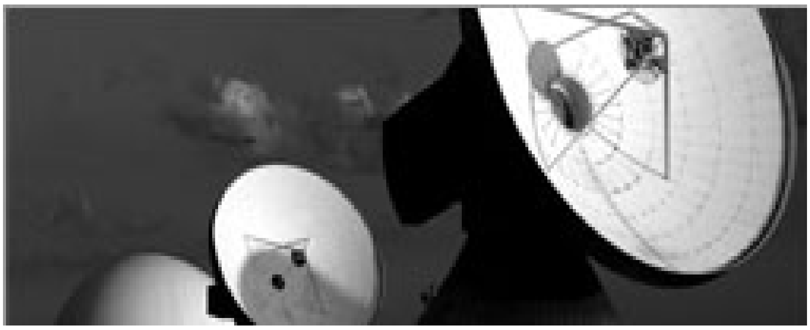
SAP Exchange Infrastructure in High Availability Environments



XI Release 2.0

Copyright

JAVA® is a registered trademark of Sun Microsystems, Inc.

J2EE™ is a registered trademark of Sun Microsystems, Inc.

JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, SAP Logo, R/2, RIVA, R/3, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP, mySAP.com, 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. MarketSet and Enterprise Buyer are jointly owned trademarks of SAP Markets and Commerce One. All other product and service names mentioned are the trademarks of their respective owners.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressively prohibited, as is any decompilation of these components.

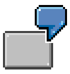
Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Documentation in the SAP Service Marketplace

You can find this documentation at the following address:
`http://service.sap.com/ha -> Media Library -> Documentation -> Misc. Documentation`

Typographic Conventions

Type Style	Represents
Example Text	Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths and options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, titles of graphics and tables
EXAMPLE TEXT	Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example, SELECT and INCLUDE.
`Example text`	Screen output. This includes file and directory names and their paths, messages, names of variables and parameters, source code as well as names of installation, upgrade and database tools.
`Example text`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.
EXAMPLE TEXT	Keys on the keyboard, for example, function keys (such as **F2**) or the ENTER key.

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Contents

Chapter 1: About This Document

Overview

This document covers the use of SAP Exchange Infrastructure Release 2.0 (XI 2.0) in a mission-critical (high availability) environment.

The concepts relevant for the high availability of XI are introduced. We discuss the high availability of application systems, the XI Integration Server and XI adapters. The corresponding configuration issues are addressed.

It is assumed that the reader is familiar with XI terminology and its technical architecture. It is also recommended that you read the relevant documentation for high availability and switchover of the SAP Web Application Server (SAP Web AS).

The current document version is 2.20.

Contents

Target Groups

This document is aimed at anyone wanting to run XI in a mission-critical environment and who therefore needs both background information and advice on how to setup and run XI to ensure optimum availability.

Note that you should be familiar with the concepts and details of HA strategies for the SAP Web Application Server (especially switchover strategies). It is also necessary that you are familiar with the general concepts of XI and its runtime environment.

This documentation is aimed at the following groups:

- Consultants
- Customers (system administrators)
- Switchover software vendors who want to adapt and test their products for compliance with XI

Other Relevant Documentation

You are strongly recommended to also refer to the following documentation:

- *BC SAP High Availability*
- *SAP R/3 in Switchover Environments*

- *SAP Web Application Server in Switchover Environments*
  (updated version of the second document with the latest SAP Web AS developments)

See the first document for a general discussion and sound introduction to the topic of high availability for the SAP Web AS. It is part of the standard SAP documentation package that is delivered together with the SAP Web AS. It deals comprehensively with safeguarding SAP Web AS system operation. You can find it at:

`http://service.sap.com/ha` → *`Media Library`* → *`Documentation`* → *`Misc. Documentation`*

For information on specific switchover products, contact your hardware and switchover software vendor. If you have any questions regarding the integration of SAP Web AS with a specific switchover product, contact the Competence Centers of SAP's hardware partners in Walldorf, Germany.

Naming Conventions

Terms & Abbreviations

The following terms and abbreviations will be used in this document:

Term	Explanation
XI	Exchange Infrastructure
HA	High Availability
Integration Server (IS)	The central XI server where XI processing (routing, mapping, queuing, conversions) takes place.
Integration Engine (IE)	The XI instance available on an SAP application system of Release 6.20 or later. The Integration Engine sends messages to, or receives messages from, the central Integration Server.
QualityOfService	XI term for describing the way to handle the transmission and processing of messages. Possible values are: BE = BestEffort (synchronous call, no transactional guarantees for transmission and processing) EO = ExactlyOnce (asynchronous call, guarantee for local transactional handling, exactly-once transmission and exactly-once processing) EOIO = ExactlyOnceInOrder (as for EO but with serialization guarantee on a given queue name).
SAP Web AS	SAP Web Application Server, latest generation SAP application server with web capabilities (after release 6.20). The term normally includes the SAP J2EE engine.
SAP J2EE Engine	J2EE part of the SAP Web AS.

AS	Application server
SID	The standard SAP 3 character system identification.
JCo / rfcengine	Java Connector, used for interconnecting JAVA software and SAP systems using RFC. JCo is available as standalone or built into J2EE (JCo is called 'rfcengine' in the J2EE context).
Gateway	Standard SAP software used for interconnections using RFC and CPIC (often also called CPIC-gateway).
Switchover Software	Software product offered by operating system vendors for customers with stringent high availability requirements. Using switchover software any component reacts automatically to failures (for example by being restarted on a standby host, executing scripts etc). SAP Web AS is designed to integrate with all common switchover products.
SPOF	Single Point of Failure. A SPOF is a component with a unique and mission-critical role within a software system. It cannot be run/held redundantly. Any failure of an SPOF directly influences the availability of the entire system.
JVM	Java virtual machine
Message Server (MS)	Standalone server component required in any standard SAP Web AS and standard SAP R/3 system for internal AppServer-to-AppServer communication and load balancing.
Enqueue Server (Enq)	Standard locking service within the SAP ABAP application server or SAP R/3 system (by default based on a specific work-process, not on a standalone server).
ICF	Internet Communication Framework is the implementation and configuration of HTTP and SMTP protocols within the SAP Web AS.
DBMS	Database Management System
NFS	Network Filesystem
A2A	Application-to-Application communication typically couples application software within a company (on the intranet).
B2B	Business-to-Business communication typically couples application software of one company (via the internet) to the (different) application software of another company.
SLD	System Landscape Directory
XI Component	The software components of the SAP Exchange Infrastructure, which can be operated independently in one package. This XI component may be operated on one cluster server at a time.
Cluster	Term for a group of servers that can be monitored and administered jointly by specific cluster software. Often

	operating systems contain cluster software by default.
(Cluster) Package	Packages are used on clusters for bundling software components and their required devices. Packages can be administered, monitored, started, stopped, relocated and so on.
Relocation / Switchover	A software component (plus access to required hardware services) is stopped on one server and restarted on a different server. Cluster software typically handles relocation on the granularity level of packages.
DMZ	Demilitarized Zone

Chapter 2: High Availability Concepts for XI

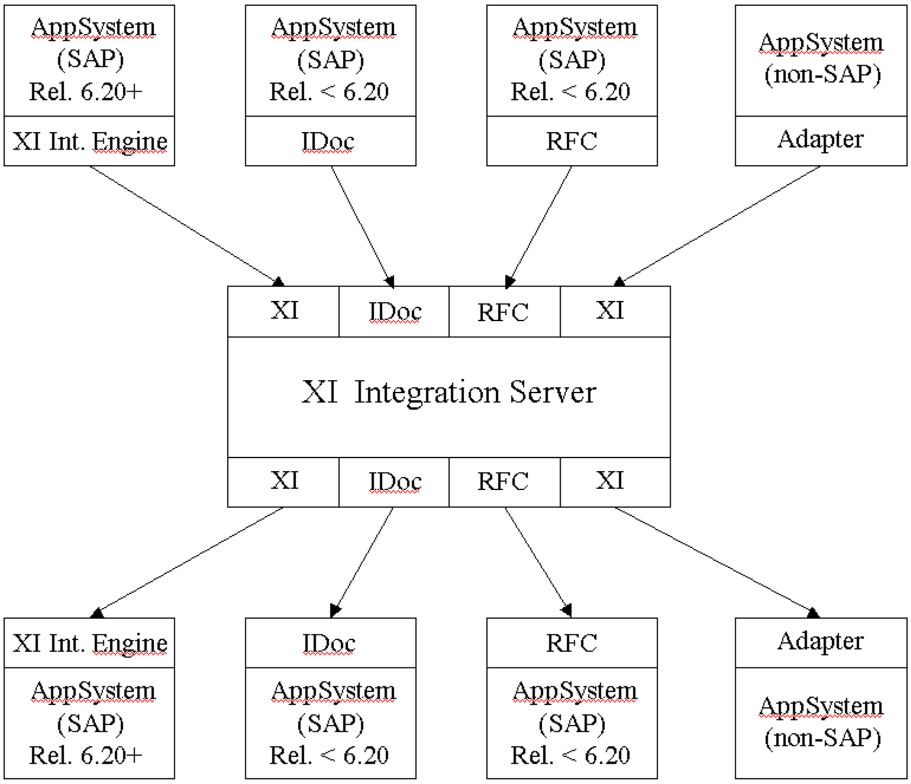
Overview

This chapter provides a high availability view of an application system landscape using XI for communication. An analysis is carried out to find single points of failure in the XI runtime focusing on the XI Integration Server. XI adapters and application systems are discussed as well. The document reviews built-in HA features of the Integration Server and provides hints for configuration (see the next chapter). The concept of switchover software in general and its use for the XI Integration Server and for application systems are briefly discussed.

Contents

Introduction

XI within the Application Landscape



The graphic above shows a schematic overview of how XI is used in an application-to-application scenario (A2A, in other words, communication happens on the Intranet) at runtime. The basic schema is very simple: A sender application system (top) talks to a central XI Integration Server (middle), which sends the messages to a receiver application system (bottom).

On a more detailed level, differences appear depending on the release and type of application systems:

- SAP systems based on SAP Web AS Release 6.20 or higher
- SAP systems with releases below 6.20
- Non-SAP systems

The latest SAP systems (based on SAP Web AS Release 6.20 or higher) usually have XI functions (ABAP proxies as well as the Integration Engine as the middleware layer) built in and thus may directly communicate with the Integration Server using XI messages (XI message format and protocol).

SAP systems based on a SAP Web AS with a release lower than 6.20 and non-SAP systems must use adapters in order to communicate with the Integration Server. In other words, adapters convert native formats and protocols to the corresponding XI conventions (and the other way around), in order to then communicate with the Integration Server.

As a result, a typical communication path is:

Sender application system
→ Sender adapter
→ Integration Server
→ Receiver adapter
→ Receiver application system

In order to discuss high availability it is important to discuss the role of all components along this communication path.

The Central Role of the XI Integration Server

It is clear from the above scheme, that the XI Integration Server – seen as a system – is a SPOF (single point of failure) for the runtime communication of any two systems that are connected across XI. This is a general notion, irrelevant of whether adapters are used or not.

The most important threat to system availability in an XI landscape is the failure of

The Integration Server system is a runtime SPOF for any XI-based communication.

A single point of failure (SPOF) is a component with a unique and mission-critical

SPOF directly influences the availability of the entire system.

As a result, the prime focus of our high availability discussion will in fact be the Integration Server. The following chapters will discuss

- SPOFs inside the XI Integration Server
- How switchover software can be used to secure Integration Server SPOFs
- High availability features already built into the Integration Server

Mission-Critical XI Adapters

Adapters may play crucial roles in XI-based scenarios. Adapters are required in communication involving systems with XI middleware not built in. It is important to note however, that the scope of adapters as SPOFs is restricted only to specific scenarios. In general, a single failing adapter will not render the entire XI-based landscape inoperable. This is in contrast to the global impact of a failing XI Integration Server.

For example, consider the XI-based communication of two legacy SAP systems, which are accessed via the RFC adapter at runtime. In this case the RFC adapter is a SPOF for any mission-critical RFC communication scenario between those two systems. However, IDoc-based communication with the same application systems (using the IDoc adapter) is not affected by the runtime availability of the RFC adapter. And of course, the failing RFC adapter will not affect independent scenarios running on the XI Integration Server.

The following chapters will discuss the different types of adapters and their type-specific resilience to failures.

Mission-Critical Application Systems

When approaching the high availability of the full application landscape, it is helpful to see application systems as independent systems. This means that

- SPOFs of application systems must be discussed
- Application system SPOFs must be secured (for example, with switchover software)
- Resilience to failures in application systems must be discussed

Even if XI is set up in a failure resilient way with optimized availability, this has no impact whatsoever on how resilient to failures a sender or receiver system will be. A mission-critical application scenario needs high availability along the complete communication path.

Application systems can be optimized in their availability by proper configuration and by implementing switchover software (very much like the XI Integration Server). For non-SAP systems please refer to the documentation of the specific product. For the case of SAP application systems please refer to the following documentation:

For a complete analysis of the SAP Web AS and more detailed discussions (down to

BC SAP High Availability

SAP R/3 in Switchover Environments

SAP Web AS in Switchover Environment.

The Runtime Role of the XI Design Time

This documentation focuses on the availability aspects of the XI runtime environment. The XI environment for design and configuration is, primarily, considered less crucial because it is not a runtime SPOF. Due to caching mechanisms implemented inside the

Integration Server the runtime availability of the design time is not strictly required. . The details of these mechanisms will be discussed below.)

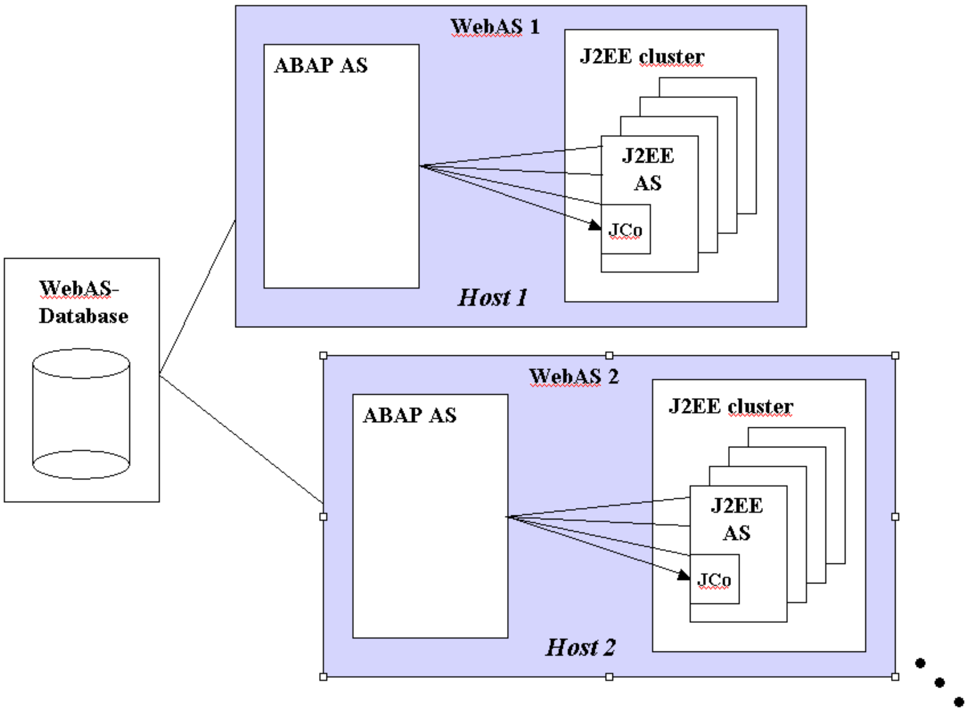
In a customer landscape, strict availability requirements may be imposed for the XI design time as well. As a result, the design time SPOFs should then be secured by appropriate mechanisms, as well.

Integration Server in Terms of High Availability

Building Blocks

There are three major building blocks, which are relevant for the runtime of a fully functional XI Integration Server:

- Database system of the SAP Web AS
- SAP Web Application Servers (ABAP-part only)
- SAP J2EE Engines



The main tasks of the ABAP part of the Integration Server are message transport inbound/outbound (using HTTP/S or adapters), message persistency and runtime processing control (both technically and logically: queuing, scheduling, service calling, routing, technical address resolution, and so on).

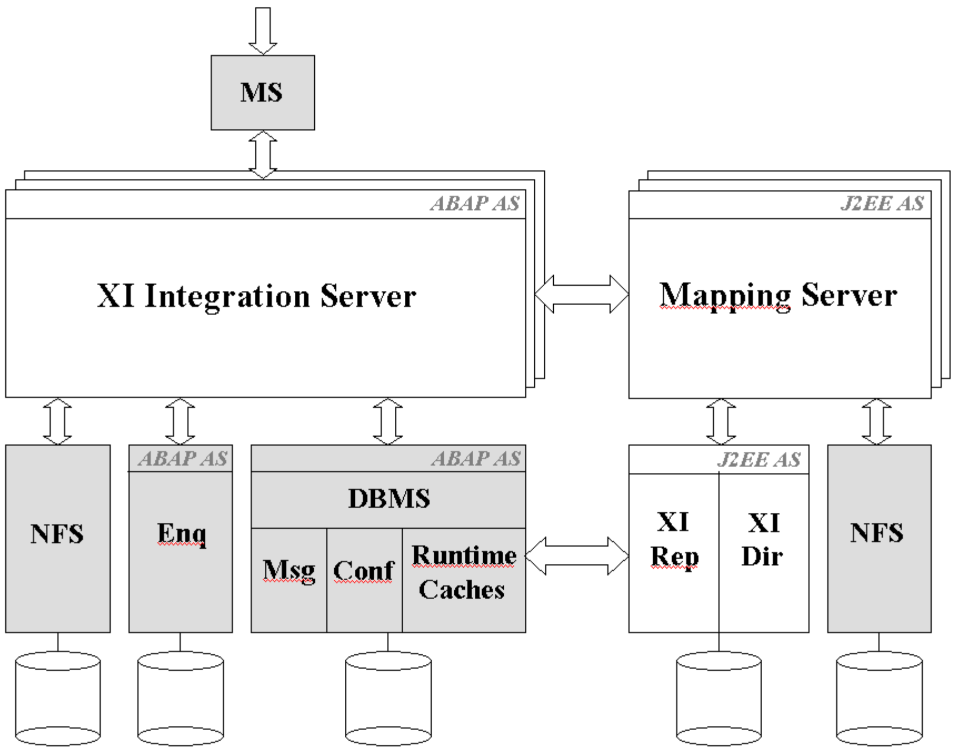
The database system can be regarded as an integral part of the SAP Web AS. Firstly, it is needed for running the SAP Web AS itself. Secondly, it provides several different kinds of database persistency needed within the XI environment (message persistency, technical configuration, logical configuration, caches and so on).

At runtime the most important task of the J2EE Engine is to provide the operational mapping environment with the possibility to implement customer-defined JAVA mappings. (Another J2EE engine also hosts the XI Integration Repository and Directory, which is serving as the master database for XI configuration data. The latter J2EE is not discussed here, although it may also be important for the runtime for cache refreshes, see below for details).

The ABAP-part and J2EE engine of a SAP Web AS interact using standard RFC calls.

Single Points of Failure in the XI Integration Server

The following graphic shows a schematic and runtime-centric overview of the XI Integration Server with its major technical building blocks and single points of failure (SPOFs marked gray).



ABAP-part of the Web AS

The XI-relevant single points of failure of the standard ABAP AS are the following:

- Database System (DBMS)
- Enqueue Server (Enq)
- Message Server (MS)
- NFS

A key use of the underlying database system is message persistency, especially in asynchronous protocols (qualities of service *ExactlyOnce* and *ExactlyOnceInOrder*) providing transactional and serialized/transactional processing. The DBMS also plays a vital role for XI's runtime caches used for holding XI configuration data (at technical and logical level) on the SAP Web AS.

The Enqueue Server is key to the transactional and serialized/transactional processing of messages. This server runs within an SAP Web AS and is based on an R/3 kernel (ABAP) work process. Thus the specific SAP Web AS with the Enqueue Server becomes a SPOF by definition.

The message server is of general importance for the low-level communication and load distribution within a multi-server SAP Web AS cluster. In an XI context it is vital for load balancing HTTP I/O onto the pool of available SAP Web ASs (note the difference in dispatching for A2A versus B2B scenarios, below).

A shared file system (NFS) is required for every standard setup of an SAP Web AS.

A system with stringent high availability requirements must ensure that the SPOFs

For a complete analysis of the SAP Web AS and more detailed discussions (down to

BC SAP High Availability

SAP R/3 in Switchover Environments

SAP Web AS in Switchover Environment.

J2EE part of the Web AS

Mapping programs and mapping XSLTs require access to an NFS-based file system (at least in XI Release 2.0). Thus NFS for the J2EE server is a SPOF for XI (as it is for the

SAP Web AS). For stringent high availability environments it is recommended that you implement switchover software to ensure that NFS remains available.

The XI Integration Server requires an operable J2EE server in order to run mapping operations. Note that XI messages, which do not require mappings, do not require the attached J2EE server either, because the call to the mapping service on the J2EE AS is skipped in this case. However, most (if not all) messages do in fact require mapping, so the J2EE AS (mapping service) availability is vital for the XI runtime as well.

The mapping service on XI is based on stateless session beans that may be run on any of the J2EE servers available in the XI Integration Server cluster. In other words at least one J2EE in the entire Integration Server system is mandatory, however, it does not matter which one is used. That is why the J2EE instance as such is not a SPOF, since J2EE instances are in general held redundantly (for scalability and in order to attain a high degree of availability).

A system with stringent high availability requirements must ensure that NFS is

In addition at least one J2EE server must remain operable for executing mappings. This can be ensured simply by running several redundant J2EE servers. It is also possible to locate at least one J2EE instance on a switchover cluster, although the J2EE-based mapping server as such is not an XI runtime SPOF.

For a complete analysis of the SAP Web AS and more detailed discussions (down to

SAP Web AS in Switchover Environment.

Additional SPOF in B2B scenarios: Web Dispatcher

As mentioned above, the message server is used to dispatch incoming HTTP calls to the pool of available SAP Web ASs. The message server employs HTTP-redirect for this task. HTTP redirect returns to the calling server and makes it re-send the HTTP request to the chosen specific SAP Web AS. This is obviously only possible if all SAP Web ASs of the XI Integration Server are visible on the network to the HTTP caller/client. Thus any solution based on the message server works on the Intranet and is feasible for A2A scenarios, only.

A different option for HTTP dispatching must be used in B2B scenarios, when HTTP callers are located on the Internet (outside a firewall). Any SAP Web AS in the XI Integration Server system is located on the Intranet (within the firewall), and thus is not visible to the outside Internet caller. Such B2B-like scenarios require a Web Dispatcher in front of the XI Integration Server in order to exploit HTTP load balancing. The Web

Dispatcher is normally situated within the demilitarized zone (DMZ) and forwards the clients' HTTP requests to the Intranet.

Much like the message server, the Web Dispatcher is a SPOF for the incoming

For background information and details about the SAP Web Dispatcher, see the

Access user menu by choosing *Help → SAP Library*:

mySAP Technology Components → SAP Web Application Server → Client/Server Technology → Architecture of the SAP Web Application Server → SAP Web Dispatcher

XI Integration Directory, Integration Repository and System Landscape Directory

The XI Integration Repository and Directory (both J2EE-based) contain information that is vital for the configuration and correct functioning of XI. All of this data is relevant at runtime when processing a message using XI. Nevertheless, the XI Integration Repository and Directory are still not SPOFs at runtime, because all relevant data is held redundantly in a cache (database table and memory-based database buffers) within the ABAP part of the XI Integration Server (see below for more details). Note that this cache completely survives SAP Web AS restarts because it resides in the Integration Server database.

System Landscape Directory

The System Landscape Directory also contains information that is vital for the correct functionality of XI. Nevertheless, it is (much like XI Directory and Repository) not a SPOF at runtime. All relevant data is held redundantly in a cache (database table and memory-based database buffers) within the ABAP part of the Integration Server (see below for more details). Again note that this cache completely survives SAP Web AS restarts because it resides in the Integration Server database.

High Availability Features in XI Integration Server

Scalability of ABAP Engine and J2EE Engine

An XI Integration Server system may consist of several instances of both ABAP and J2EE engines. It is a well-known benefit of this setup that performance and total throughput increases with the number of server instances (primarily a scaling option). In fact, the SAP Web AS has proved to scale in approximately linear fashion over a wide range.

In addition, scalability also increases the high availability of XI, because ABAP and J2EE engines are held in several redundant instances. The system can both tolerate and automatically compensate failures of a portion of its server instances. This notion is valid as long as the relevant SPOFs, one ABAP engine and one J2EE engine continue to function, of course (the latter point is addressed by switchover software as outlined below).

Transactional Message Transmission & Processing

Messages with QualityOfService *ExactlyOnce* (EO) and *ExactlyOnceInOrder* (EOIO) are handled in a transactional manner within the XI system. Transaction data and states are recorded on the underlying database system. That is why both data and states are resilient to failures in general (because DBMS mechanisms are used).

Message Transmission

Message transmission from one XI instance to another is 'transactional': The sender continues to re-send a message until it gets a positive notice that the message has in fact been persisted on the receiver. In other words, as long as network errors and such prevent the message from arriving at its destination, the message will continuously be sent by the XI sender (see *Automatic Retry* below).

On the other hand, an XI receiver evaluates the unique message ID in order to recognize any duplicate message being re-sent into it. Thus, the re-send action will be tolerated and positive receipt of the message will be acknowledged (leading the sender to stop re-sending this message). In any case it will be ensured that a message is persisted only once (processed and/or forwarded).

The combination of the two features above ensures an ExactlyOnce transmission of message data in XI. Therefore it is called the **XI ExactlyOnce Protocol**. Both message data and message state are held in the XI message persistency within the SAP Web AS database system. That is why both data and states are resilient to failures.

Message Processing

Message processing within an XI server (both Integration Server and Integration Engine) is controlled using message ID and standard SAP transactions (ABAP-based).

While being processed, any message goes through well-defined state transitions on the database. In case of failures, it is ensured that message data and states (plus application data on the Integration Engine) are consistent and processing can be started from scratch again. This resilience to failures is based on the resilience of the DB system, of course. Note that the use of a database system becomes one of the serious benefits of XI in this respect.

Messages entering XI through adapters will ensure this behavior if the processing mode (QualityOfService) is EO or EOIO (provided the specific adapter is explicitly capable of EO/EOIO handling).

Automatic Retry

XI runtime contains the concept of a 'retriable' error state for a message. Retriable errors are error situations with a transient cause, for example, network failure, transient unavailability of a target system, but also if application data is transiently locked for update by another user. Messages in retriable error state are automatically re-processed after some pre-defined period of time (normally every 2 minutes with 10 retries in total, if not configured differently).

This feature ensures that transient unavailabilities of systems or specific services can be tolerated and handled transparently, thus increasing the failure resilience of the entire XI system. Short outages of networks and application systems can be tolerated without the necessity of manual interaction.

Note however, that after a pre-defined number of retries (retry timeout), the message

message must be re-started manually (administrator) or by using dedicated batch jobs.

HTTP Dispatching and Effects of Failures

XI Integration Server/Engine is capable of parallelizing HTTP call processing on several servers using either the message server or the Web Dispatcher. These processes either redirect (message server) or forward (Web Dispatcher) HTTP calls to a specific SAP Web AS within the cluster for XI processing. Every Integration Server/Engine instance (every SAP Web AS) itself persists the message data and its state in the underlying database (shared on the Integration Server system).

Sudden failure of one server instance (SAP Web AS), does not severely affect the availability of the system:

- If one SAP Web AS fails, the dispatcher (message server or Web Dispatcher) will be notified immediately. Thus further calls will not be forwarded or redirected to the failed server. Only the calls presently entering the failing server will hit the failure.
- In case of synchronous calls (QualityOfSevice is *BestEffort*) the application currently connected to the failing Integration Server instance will receive a runtime error. However, the application may re-send the call to XI immediately, because another Integration Server instance will be available to handle the request. Load-balancing will ensure that an alternative server is used when the application-based retry call is made.
- In case of asynchronous calls (QualityOfSevice is *ExactlyOnce* and *ExactlyOnceInOrder*) the application is completely decoupled from errors happening on the Integration Server or at message transmission time. All XI transmission partners (for example an Integration Engine sending to an Integration Server) contain a secure transmission protocol (ExactlyOnce transmission protocol) for making server failures transparent (see the section on transactional message transmission above). This includes automatic re-transmission on the sender and duplicate message recognition on the receiver. Two cases may be distinguished here:

- If an asynchronous message transmission fails and the data has not yet been persisted on the receiver, the message will be re-sent by the caller. Thus it will finally be received (and persisted and scheduled for execution) by the target server. The sender will finally be informed by the receiver that the message has arrived.
- If an incoming asynchronous message transmission fails and the data has already been entered successfully on the receiver's database, the data will definitely be processed on the receiver system. If the receiving server instance has failed in the meanwhile, another instance of the Integration Server cluster will automatically take over processing (irrespective of which server actually persisted the data initially). If the sender re-sends the message, the receiver recognizes the duplicate and tells the sender to stop re-sending.

Action Required

The message server (or Web Dispatcher) must be configured to represent the address-URL of the XI Integration Server/Engine to the outside world.

It is generally recommended that you set up XI Integration Servers/Engines with

switchover software and increases the failure resilience of the XI system.

RFC Dispatching

Normal XI message traffic enters the Integration Server using the HTTP protocol. However, if the IDoc and RFC adapter is used, message data enters the Integration Server by using the standard RFC protocol.

Note that load balancing is also available for the RFC protocol. The Web AS message server is used as the call dispatcher (as in the HTTP case). RFC load balancing offers two major benefits for the XI system:

- Improved scalability - calls are parallelized and forwarded to several different SAP Web AS instances.
- Improved availability - any SAP Web AS of the XI Integration server can be the target of the incoming call. It is not required to have one specific Web AS functional, any Web AS instance will do.

With RFC load balancing activated, sudden failure of one SAP Web AS, will not affect the accessibility of the XI system through the RFC-based adapters.

Action Required

In order to increase the accessibility of XI through the IDoc or RFC adapter, you should set up the RFC destinations addressing the Integration Server accordingly (configure load balancing in the RFC destinations of the caller systems). This means that the message server address must be used to represent the target of the RFC destination for the RFC or IDoc adapter.

It is generally recommended that you set up the RFC destination to the XI Integration

This feature is independent of the use of switchover software and increases the failure resilience of the XI system.

Caching for XI Integration Repository and Directory

The primary data for configuring the XI environment is located in the XI Integration Repository and Directory (both are J2EE based). Of course, these databases are ultimately required to configure and operate the XI runtime. However, they are not explicitly required to function without interruption at runtime. Strictly speaking, the XI Integration Repository and Directory are not XI runtime SPOFs .

At XI runtime the relevant Repository and Directory data is cached in the memory and database of the ABAP part of XI. This feature ensures the following:

- Runtime-relevant data can be accessed quickly and reliably within the ABAP part of the Integration Server.
- Transient unavailability of the XI Integration Repository and Directory (for example unavailability of the relevant J2EE engine) does not directly affect message processing in the Integration Server runtime.

If any changes are made to the configuration of the XI Integration Repository and Directory, the runtime caches are refreshed automatically. Note however that, this cache refresh mechanism means that there may be a short time delay between saving a change in XI tools and their active use at runtime. While caches are still being refreshed, the previous settings are used. The XI runtime system will also use the previously valid configurations and programs as long as changed ones cannot be loaded into the runtime cache, for example in case of unavailability of the Integration Repository or Directory (needed for refresh).

As soon as the XI Integration Repository and Directory become accessible again, all caches will be refreshed automatically.

If a message has been processed to a final state with pre-change configuration data

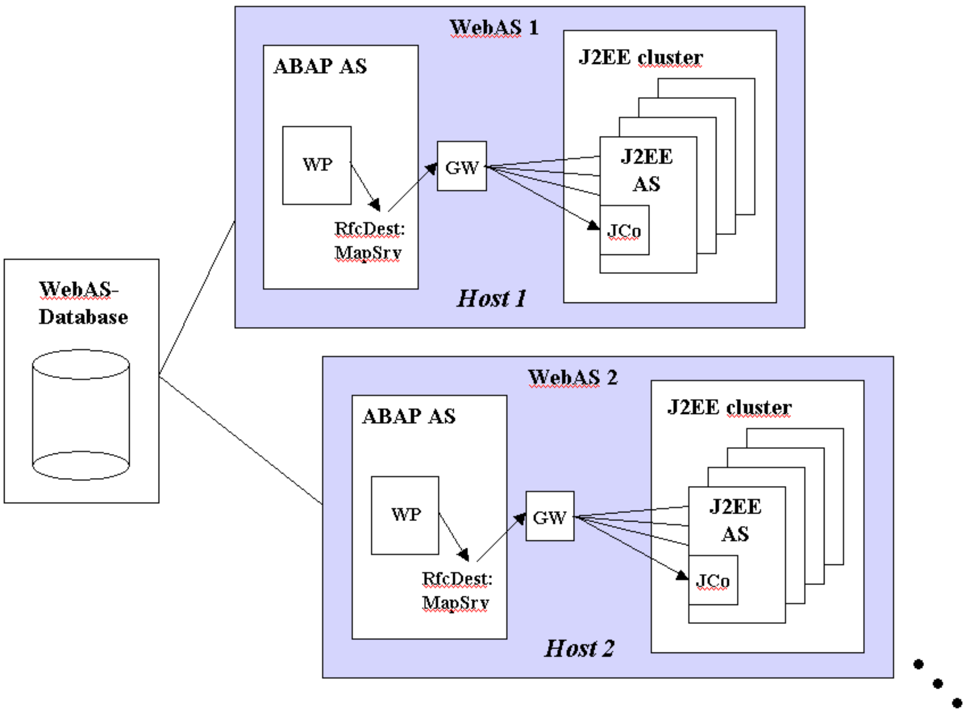
will **not** be reprocessed with new configuration data when the cache becomes updated.

This behavior cannot be avoided, because the data may already have been entered and committed in the target application system (data cannot be rolled back anymore in the application system).

Availability of the J2EE Mapping Server

The J2EE-based mapping server is not an XI runtime SPOF, because it can be held redundantly. It is generally recommended to run several J2EE engines concurrently with the Web AS on every single host machine of the Integration Server system. This will be handled by standard installation already, however.

If a single J2EE engine or virtual machine fails, only the calls that are currently running on this J2EE engine will be aborted locally (and automatically retried by the Web AS in the case of asynchronous calls). The (CPIC) gateway (which acts as a dispatcher to the local J2EE cluster) will recognize the failure and will not dispatch any further calls to the failed J2EE until the server is functioning again and rejoins the J2EE cluster. Any follow-up requests (even from the same SAP Web AS as before) will be dispatched to a different J2EE server.



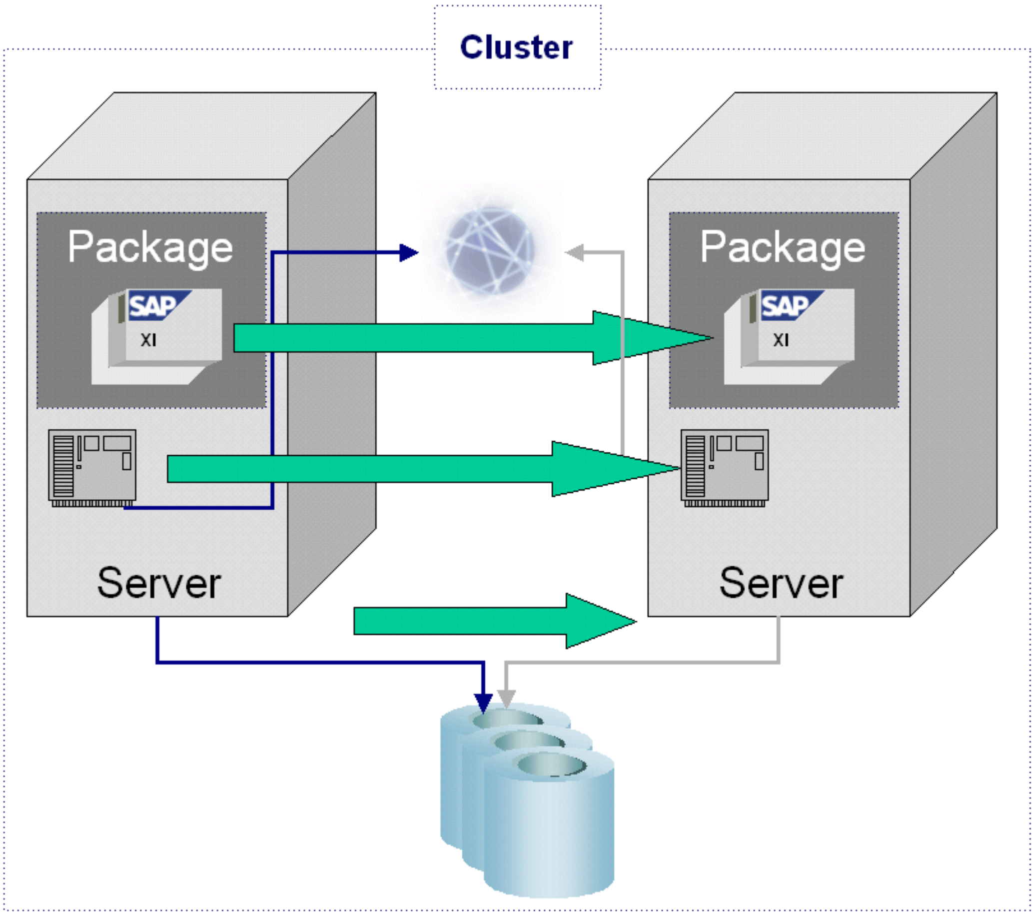
Switchover Environments for XI

Principles of Cluster and Switchover Technology

Cluster Technology and Packages

Modern operating systems implement *cluster technology* for use in high availability environments. Crucial software components are bundled into *packages*, which can easily be managed by cluster management software (monitored, stopped, started, relocated from one cluster server to another and so on).

Packages are usually those software units that are relocated from one server to another for failure reasons (for example, hardware failure) in a high availability setup. Of course, several other resources (such as disks, network interfaces and so on) need to participate in this process of relocation.



Switchover Technology

The basic idea of switchover technology is to restart SPOF software components on a standby server if the primary server for this function has failed. Both the primary and the standby server reside inside a cluster, which is monitored and managed by cluster software. In general, only SPOF components (in other words, the crucial packages) need to be secured by switchover software (see above for the list of XI SPOFs). Non-SPOF components are made failure-resilient simply by running several redundant instances of them.

For a detailed introduction into the topic of switchover technology see the following

BC SAP High Availability

SAP R/3 in Switchover Environments.

SAP Web AS in Switchover Environments

Addressing

As software components are mostly interlinked across the network, it is necessary for those still functioning to re-access the failed SPOF component as soon as possible. In order to make the failure transparent for network addressing, so-called virtual IP addresses (and corresponding hostnames) are used in such environments. Modern operating systems normally allow the administrator to attach virtual IPs to cluster packages and thus to software units. It is helpful to think of virtual IP addresses as tightly attached to their SPOF components. A SPOF software component and its virtual IP address are switched jointly from server to server in the event of a switchover.

For this reason, it is important to exclusively use virtual addressing for access to SPOF components. Only in this case does addressing remain constant for the surviving environment. Most typical problems in switchover systems are solved by proper configuration of network addresses and the hostnames that address them.

Concepts for Switchover of the Integration Server

As outlined above, the following XI software components are runtime SPOFs:

- Database Management System
- Enqueue Server
- Message Server
- NFS

All of these SPOFs are vital for XI to function properly at runtime. If just one of these components is unavailable then the XI Integration Server is rendered inoperable. If stringent high availability requirements are made, it is recommended that you implement switchover software products to secure the availability of the DBMS, enqueue server, message server and NFS within the ABAP part of XI.

Note that NFS is also a SPOF for the J2EE part of XI. So its availability must be secured by switchover software, as well.

There are further important components of XI, especially tools and databases needed for configuration. Strictly speaking, these components are <u>not</u> SPOFs in a runtime sense. In fact, they can either be held redundantly (load balancing switch to a second instance) or a short unavailability of them (for example a configuration tool working on an XI Integration Repository/Directory) can be tolerated by the system (for example because the configuration-settings are cached).

Nevertheless, it is recommended to run the XI Integration Repository and Directory also on a J2EE server with switchover software installed (at least to secure the database against failure). This will ensure a high degree of availability for these important components. Thus, configuration entries and programs can be changed, viewed and refreshed into the XI runtime caches most of the time.

The major SPOF components in the XI runtime are identical to those inherent in the

for more detailed discussions (down to configuration level) see the following documents:

BC SAP High Availability

SAP R/3 in Switchover Environments

SAP Web AS in Switchover Environment.

NFS is a runtime SPOF for both the ABAP part and J2EE part of XI.

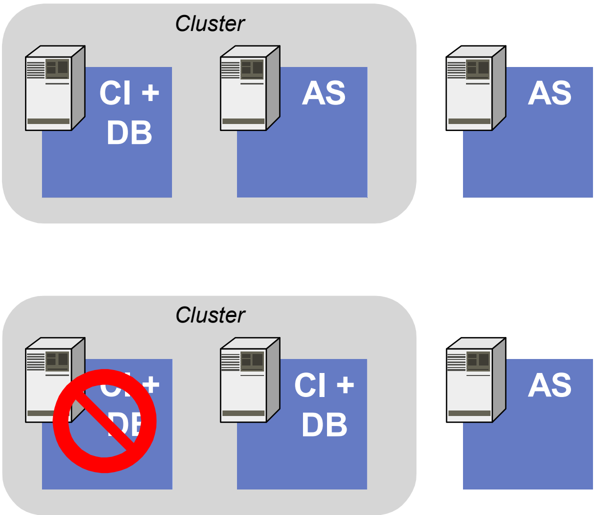
Example: Simple Option for a Switchover Cluster

There is a variety of ways of 'packaging' XI system SPOFs into a system setup of clustered application servers.

All relevant setup and distribution options and their pros and cons are discussed in the

more complex switchover cluster setups for a XI landscape.

The setup chosen depends on several further technical system decisions that should be made before the system XI is set up in a high availability environment (for example number of application servers, use of a standalone database server, and so on).



For example, consider one of the simplest setup options (mostly used on UNIX servers), which runs an SAP Web AS jointly with the enqueue server and message server on a single hardware node (this is called a 'central instance' or CI). In addition the CI node also carries the database system and NFS (in short: CI+DB). An additional SAP Web AS (AS) is included in this system for performance scaling in normal operation. For simplicity J2EE components and NFS are not explicitly shown in this graphic (NFS is located on the CI, every server contains a J2EE engine).

In a crash situation (for example hardware failure of the CI+DB server) the second application server is used as a hot standby for switchover. Thus, two nodes are bundled into a switchover cluster (CI+DB server plus standby AS). In the case of a failure, the switchover product will startup all relevant SPOF components on the standby server and thus make the entire system functional again.

This setup option is depicted above in normal operation (upper part) and after failure of the critical node (lower part). Note that any number of further application servers (AS) can be added to this cluster solely for scalability reasons (additional AS to the right of the cluster).

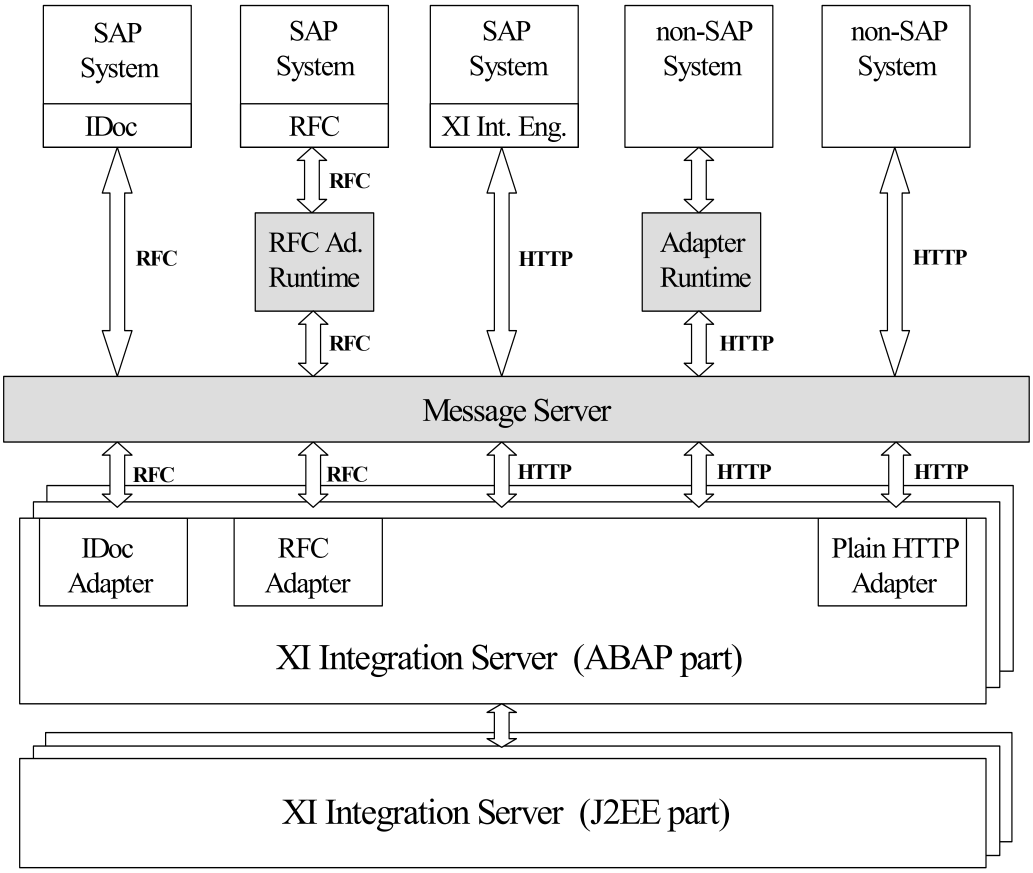
Database versus NFS Availability for J2EE

Database availability is vital for the ABAP part of XI to function correctly. Note however, that at runtime, the J2EE part (mapping server) does not require access to the database system (in XI Release 2.0).

Mapping programs and XSLT are not held in the database system (in XI Release 2.0), but on the shared file system. This means, that for the runtime of J2EE in XI the NFS component is vitally important and must be secured by switchover software.

XI Communication Involving Adapters

Adapter Architecture Basics



Adapter Types

From an architectural point of view there are three different types of adapters in XI systems:

- Standalone adapters (XI Adapter Engine)
- Adapters built into the ABAP part of the XI Integration Server system
- Adapters with an IS-based part and a standalone part (inside the Adapter Engine)

Of course, there is the additional option to have no adapter at all. In other words both partners communicate by natively exchanging XI messages across HTTP.

Standalone Adapters

Standalone adapters are started as standalone executables (on the Adapter Engine) outside the Integration Server system (see the second from right in the general scheme given in the graphic above). Examples of standalone adapters in this sense are:

- File adapter
- JMS adapter
- JDBC adapter

Standalone adapters communicate with the Integration Server by using HTTP. It is possible to use call dispatching from the adapter to the different Integration Server instances (Web AS instances inside the Integration Server system) by addressing the standard message server of the Integration Server.

IS-based Adapters

IS-based adapters are run inside the Integration Server in a work process (see the leftmost and rightmost adapters in the general scheme given in the graphic above). The work process may run on any of the SAP Web AS instances inside the Integration Server system. Examples of IS-based adapters are:

- IDoc adapter
- Plain (HTTP) adapter

External application systems (SAP or non-SAP) directly communicate with the XI Integration Server. They use either plain HTTP or RFC (for IDoc transmission) as technical protocols. By addressing the message server both RFC (IDoc transmission) and HTTP protocols allow call dispatching to any of the Web AS instances inside the Integration Server system.

Mixed-Type Adapters

The mixed type technically consists of two software components, one is a standalone executable (on the Adapter Engine), the other is run in a work process on any Web AS of the Integration Server system (see the second from left in the general scheme given in the graphic above). The only adapter of this type currently is the:

- RFC adapter

An external SAP application system wants to communicate with the XI Integration Server using normal RFC application calls. By addressing the standard message server the RFC protocol allows call dispatching to the any Web AS instances inside the Integration Server system.

No Adapters, XI-to-XI communication

In this case both communication partners have built-in XI functions with respect to both the message format and the message exchange protocol (see the middle row in the general scheme given in the graphic above). This scheme applies for the following communication setups:

- Integration Engine to Integration Server

* Integration Server to Integration Engine

The technical transport protocol is HTTP. Again, call dispatching can be used by addressing the message server of the opposite XI system (which may be either an Integration Server or an application system with a built-in Integration Engine).

Adapter Availability

Availability Issues for Adapters

In general, adapters are required to convert data from non-XI format to XI format and for handling the communication inbound or outbound with the XI Integration Server. Therefore, in terms of high availability, the following aspects are important for the runtime of XI adapters:

* Adapters must be available for execution.
* Adapters need to address the Integration Server.
* Adapters need to be addressed by the Integration Server.

Adapter Resilience to Failures

Due to their architectural differences, adapters have different degrees of resilience to failures:

* **Standalone adapters** (adapters for file, JMS, JDBC) need to be restarted explicitly in case of failures (hardware or software failure). This may be done by built-in mechanisms (usually by starting the adapter as a Windows service or UNIX daemon process, see adapter documentation for details on this topic) or using switchover software.
* **IS-based adapters** (adapters for IDoc, plain HTTP) are automatically available within any of the Web AS in the XI Integration Server system. Adapter restart is not necessary as long as at least one Web AS remains functioning. Adapter redundancy is inherited from the Web AS. No further mechanisms (besides securing the Integration Server itself, switchover software) are needed to safeguard these adapters in a high availability environment.
* **Mixed-type adapters** (RFC adapter) inherit the properties of standalone adapters and their standalone part requires explicit restarts in case of failure. Again, this may be done by built-in mechanisms (usually by starting the adapter as a Windows service or UNIX daemon process, see adapter documentation for details) or using switchover software.
* In the case of **XI-to-XI communication** (this refers to the case ‘no adapter used’) the Integration Server communicates with an external Integration Engine, which is located in an SAP application system of release 6.20 or higher. In terms of failure resilience an Integration Engine closely corresponds to the case of IS-based adapters. Restart is handled by the Web AS architecture. If the application system is secured by implementing switchover software, this will automatically also secure the built-in Integration Engine.

Redundancy and Addressing Issues for Adapters

Redundancy is one of the most important high availability features of software systems. It means that a software component is held in several instances, and (if possible) it is transparent to the user or the runtime system which instance out of several is actually used to perform a specified task.

As an example consider the SAP system architecture, which enables you to run several Web AS instances in one system. External calls (for protocols RFC and HTTP) address a dispatching component (standard message server), which distributes the call on one of the Web AS instances. In case one Web AS fails, new calls are distributed to the remaining instances.

Redundancy offers immense benefits in terms of high availability. However, note that incoming calls always expect to see the dispatching element (message server) under a constant address (see the discussion of transparent addressing in the sections above). Thus the maximal benefit of redundancy can only be expected if constant and transparent addressing is ensured by the system architecture as well.

Message server dispatching is only relevant for the incoming message traffic. In other

system with all of its SAP Web AS instances.

If messages are sent out from the Integration Server (IS) to receiver systems, the IS message server is irrelevant for the outbound communication because it is not used at runtime.

However in this case, the message server of the receiver application system should be addressed in order to use call dispatching on the (inbound) receiver side.

Due to their architectural differences, adapters pose different problems for redundancy and addressing in an high availability environment.

Addressing - Application System → Adapters → Integration Server

Application systems see redundant instances of an adapter as separate target systems. That is why the redundancy of the adapters currently cannot be hidden from the application systems. A failing adapter shuts down one of the channels to the Integration Server. Other channels can only be used by explicitly using a different destination or address.

If an adapter needs to address the Integration Server, the Integration Server system is always available under a constant address - even in case of failures (when employing switchover software). The way to achieve this sort of behavior is to address the message server of the Integration Server and use load balancing (as discussed above). This dispatching / load balancing is available for both technical access protocols (see RFC and HTTP in the graphic above). Consequently it is available (currently) for all XI adapters.

Of course, you need to secure the Integration Server including all of its SPOFs (especially the message server in this context) by switchover software.

Addressing - Integration Server → Adapter → Application System

The XI Integration Server must in turn address an adapter under a constant address - even in case of failures. Several cases need to be distinguished here:

- **Standalone adapters** can be held in redundant instances in two different ways.

  1. An Adapter Engine may host several adapter threads inside of one executable process. Automatic dispatching takes place between these threads. Only one address of the Adapter Engine is required for access from the Integration Server point of view.

  2. Two Adapter Engines can be installed. This option offers more resilience to failures. However, currently the address of both Engines must then explicitly be known to the Integration Server. There is no automatic dispatching facility in this case.

     In order to secure an important adapter it is also possible to use switchover software.

     If one or several standalone adapters are used (in other words, no switchover software implemented), the individual instances may fail and thus may not be addressable for an unknown period of time. However, automatic retry in the Integration Server will try to resend XI messages into these adapters (with timeouts).

     In any of the above cases, an adapter will be able to address the application system, if the application system is made available under a constant address.

- **IS-based adapters** need not be addressed from the Integration Server to the outside world, because they are residing inside the Integration Server system already.

  However, the adapters themselves should use central and transparent addressing points for their underlying application systems. This means the IDoc receiver should be addressed via a destination employing load balancing (address the standard message server of the target SAP application system), the HTTP receiver should – if possible – reside behind a constant dispatching address.

- **Mixed-type adapters** have the same properties as standalone adapters, because most of their properties are inherited.

- In the case of **XI-to-XI communication** the Integration Engine in an application system (release 6.20 and higher) should be addressed using dispatching and load balancing to its standard message server. The message server of the target application system should be secured by switchover software (much like the message server of the Integration Server system). Securing an SAP application system with switchover software will automatically secure the built-in Integration Engine, because the runtime SPOFs in Integration Server and Integration Engine are identical.

Chapter 3: XI Installation in Switchover Environments

Overview

This section gives an overview of XI installation and additional modification tasks that have to be performed in order to use XI in a switchover environment. In order to provide a reasonable template for customer setups, a simple switchover environment of two servers is described here. This setup can be tested within simple XI switchover scenarios, which are outlined in the next chapter. The setup may also serve as a guideline for designing and installing more complex landscapes. The topic of address changes (physical versus virtual hostnames) is discussed in detail.

Contents

Conventions for Installation

Setup Philosophy

Several options exist to setup XI on a high availability cluster environment. Each individual setup situation might imply specific constraints on how to package XI runtime SPOFs and on how to distribute these packages across the cluster. Thus the decision on how to proceed here might be rather complex and might evolve over time at a customer site. This document can only provide a reasonable and simple template for starting a setup that should meet all basic requirements of a XI high availability environment.

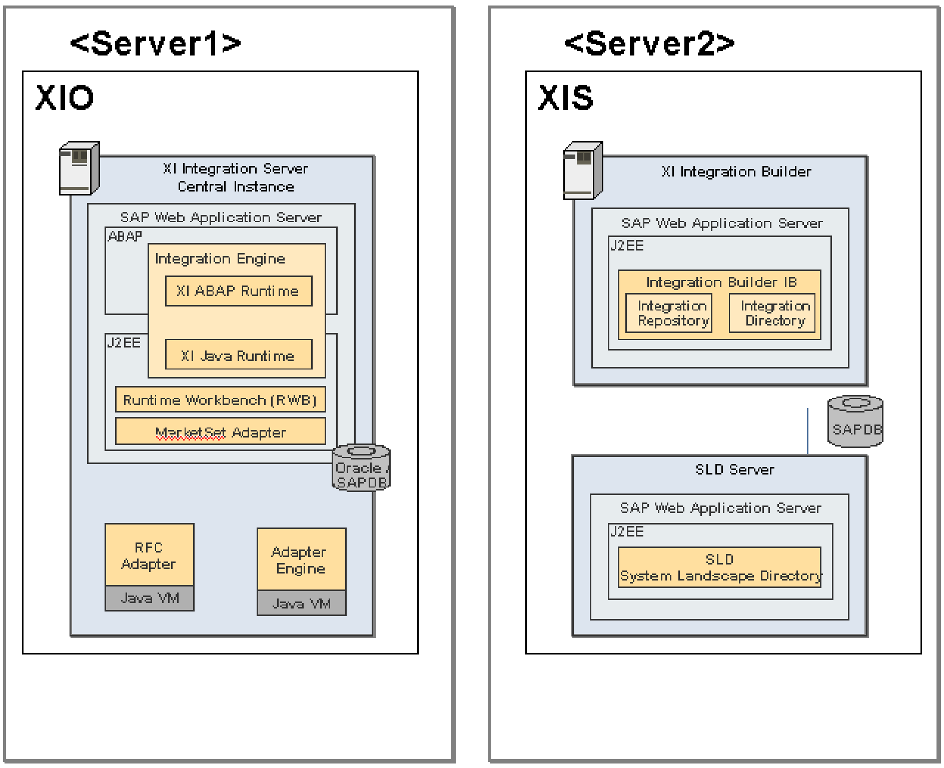
Note that the XI configuration software is not considered a runtime SPOF for XI (due to caching on the IS, see discussion above). That is why the Integration Builder (XI Repository and XI Directory) and SLD are not included in the switchover cluster setup described below.

The Template Setup

The following description of a template XI installation for switchover addressing is based on a cluster of two servers (<Server1>, <Server2>) for distributing the necessary XI components. The following assumptions are made:

- **XI Integration Server**
  installed on a cluster consisting of <Server1> and <Server2> with the SID = XIO
- **XI Integration Builder**
  installed on <Server2> with the SID = XIS
- **SLD Server**
  resides on <Server2>
- **Standalone Adapter Engine**
  resides on <Server2> (used as a sender and receiver system)

See the graphics below for an overview on the setup which is described and used throughout this document. During normal operation the XIO system is hosted on <server1>. After a switchover <server2> will host the XIO system as well.



XI Installation

Web AS Installation Issues in Switchover Environments

This documentation assumes that the SAP Web AS system underlying the Integration Server has been setup with respect to addressing requirements in switchover environments.

For more detailed discussions (down to configuration level) see the following

SAP R/3 in Switchover Environments

SAP Web AS in Switchover Environment.

Installing the Integration Server

The Integration Server package is installed with the SID XIO. This component is an SAP Web AS 6.20 with Unicode and an Oracle Database. The instance number is assumed to be 20 in our examples.

Hostname Conventions

This document is based on the assumption that the installation is done using the physical hostname. The conversion to the virtual hostname was performed in a separate step. Please proceed as follows:

- Install with the physical hostname of <server1>.
- In order to allow for transparent XI addressing in a switchover environment, the physical hostnames must be substituted by the appropriate <virtual hostname> afterwards. For details see section *Changes for high availability*.

Documentation Used

- Master Guide – SAP Exchange Infrastructure 2.0
- Installation Guide – Software on UNIX: OS Dependencies – Web AS 6.20
- Installation Guide – Unicode SAP Web Application Server 6.20 on UNIX: Oracle
- Installation Guide – SAP Exchange Infrastructure 2.0

Installation

The installation should proceed as follows:

1. Install the System XIO on server1 according the Web AS installation guide
2. Install the SAP XI Add-On
3. Create client 100 for the SAP Exchange Infrastructure

4. Update the main components as described under
`http://service.sap.com/patches`
→ *SAP EXCHANGE INFRASTRUCTURE*
→ *SAP EXCHANGE INFRASTRUCTURE 2.0*

Installing the Integration Builder

The Integration Builder package is installed with the SID XIS. This component is an SAP Web AS 6.20 with Unicode and SAP DB. The instance number is assumed to be 30 in our examples.

Documentation used

- Master Guide – SAP Exchange Infrastructure 2.0
- Installation Guide – Software on UNIX: OS Dependencies – Web AS 6.20
- Installation Guide – Unicode SAP Web Application Server 6.20 on UNIX: SAP DB
- Installation Guide – SAP Exchange Infrastructure 2.0

Installation

The installation should proceed as follows:

1. Install the System XIS on server2 according the Web AS installation guide.
2. Install the SAP XI Add On
3. Create client 100 for the SAP Exchange Infrastructure
4. Update the main components as described under
`http://service.sap.com/patches`
→ *SAP EXCHANGE INFRASTRUCTURE*
→ *SAP EXCHANGE INFRASTRUCTURE 2.0*

Basic XI Configuration

Documentation used

- Installation Guide – SAP Exchange Infrastructure 2.0
- Configuration Guide – SAP Exchange Infrastructure 2.0

Please refer to this documentation for further details.

SLD Configuration

The SLD configuration is the preparation for the test scenarios.

The SLD is called at `http://<server2>:53000:/SLD`.

Technical Landscape

Define in the SLD 3 new technical systems: XIO, SND and RCV

1.) XIO is an SAP Technical System
an SAP Web AS 6.20
on <server1>
with the Instance Number 20
and the client 100

2.) SND is an SAP Technical System
Installed Product: SAP SRM E-PROC 3.0 (EBP 4.0), for example
on <hostname>
with the Instance Number 00
and the client 300.

3.) RCV is an SAP Technical System
Installed Product: SAP SRM E-PROC 3.0 (EBP 4.0). for example
on <hostname>
with the Instance Number 00
and the client 200.

XI Landscape

Define in the SLD 3 new business systems:

1.) "IntegrationServer":
SAP Business System
XIO on <server1>
client 100 of XIO
SAP Web AS 6.20
Business System Role: Integration Server
Pipeline URL:
`http:/<server1>:52000/sap/xi/engine/entry?action=execute`

2.) "SND_300"
Choose these entries:
SAP Business System
SND on <hostname>
client 300 of SND
SAP Web AS 6.20
Business System Role: Application System
Related Integration Server: XIO_100

3.) "RCV_200"
Choose these entries:
SAP Business System
RCV on <hostname>
client 200 of RCV
SAP Web AS 6.20
Business System Role: Application System
Related Integration Server: XIO_100

Installation RFC Adapter

The configuration of the RFC Adapter has to be performed as described in the SAP Exchange Infrastructure Configuration Guide. It is assumed that the adapter is installed on <server1> in the following.

Installation Standalone Adapter Engine

On <server2> a standalone Adapter Engine has to be installed and configured. This Adapter Engine is used for the Test scenario 1.

This Adapter Engine will be delivered with a preconfiguration of scenarios used for the scenarios described here.

- Obtain SAPXICONS01_3-10002050.SCA from the SAP Service Marketplace at http://service.sap.com/patches
  → SAP Exchange Infrastructure 2.0
    → SAP Exchange Infrastructure 2.0
      → Binary Patches
        → XI Connectivity SE 2.0
          → OS independent.
- Extract the file `TechnicalAdapters.sda` from SAPXICONS01_3-10002050.SCA
- Extract the files in `TechnicalAdapters.sda` into the directory

  `/usr/sap/adapter directory` (Unix)

  `<drive:>\usr\sap\adapter` (Windows)

Remember to copy the file `SERVLET.JAR` into the path `<drive>:\usr\sap\adapter\tech_adapter`.

This file can be found on the CD SAP Web AS Release 6.20, Java Development environment (CD 51017553) under `Java\JBA\LIB\EXT`.

The installation should be performed according to the provided online documentation under *SAP Exchange Infrastructure* → *Runtime* → *Adapters* →*Adapter Engine*.

The adapter engine can be started with the command

```
/usr/sap/adapter/run_adapter.sh      (Unix)
C:\usr\sap\adapter\run_adapter.bat  (Windows)
```

Afterwards the administration tasks can be performed at `http://<server2>:8210`.

If you need to change the GUI admin port (8210), edit the value of the port in the file

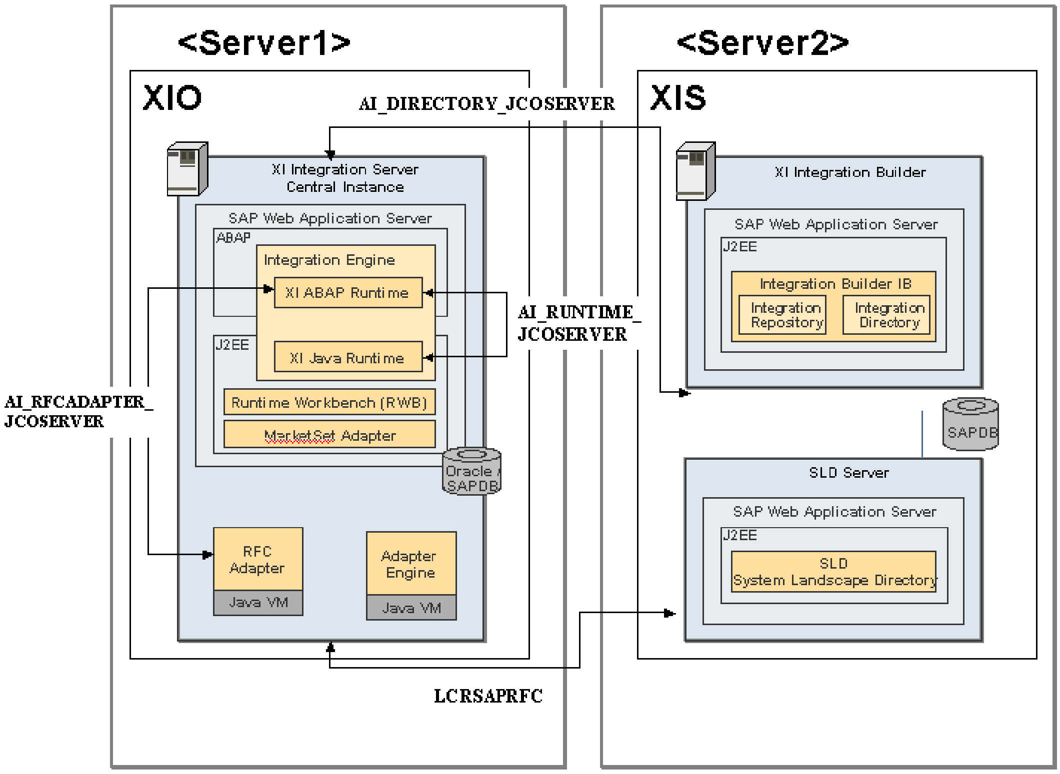
Changes Required for Switchover Environments

Connections from/to/within the Integration Server

The graphic below shows the following:

- Connections between the XI Integration Server XIO and the other components of the system
- Connections required for internal communication inside the Integration Server

All of the addressing items shown must remain constant and be available transparently after a switchover. This makes sure that the Integration Server can continue fulfilling its tasks without the need to change any address settings.



Details of the address settings will be discussed below.

Hardcopies for the required modification tasks can be found in the appendix.

Changes in:	**Integration Server XIO**	**Integration Builder XIS**	**RFC Adapter**	**Standalone Adaper Engine**
WebAS: RFC-Destination	AI_DIRECTORY_JCOSERVER AI_RUNTIME_JCOSERVER			
J2EE: rfcengine	AI_RUNTIME_XIO	AI_ DIRECTORY_XIO		
DBPool	XIRun			
security	SAPIntegration			
Files:	dbconnect.properties	dbconnect.properties	properties file	SLDaccessor.properties
Changes in:	**Exchange Profile:**	**SLD:**		
	(under connection)	*(under Content Maintenance)*		
	com.sap.aii.connect.integrationserver.name	class: BC_SYSTEM		
	com.sap.aii.connect.rwb.name	class: BC Client		
		class: BC Application Server		
		class: BC Message Server		
		class: Http Service Port		

Host Names

The physical host names must now be substituted by their respective virtual host names. For the scope of this description the XIO server will be referenced by <server1>, the XIS server will be referenced by <server2> in the following.

Changes to the Integration Server

The virtual hostname of the cluster package containing the Integration Server XIO

RFC Destinations (Web AS XIO)

Change the **gateway host** from <server1> to <virtual_1> for the following RFC destinations:

- AI_DIRECTORY_JCOSERVER
- AI_RUNTIME_JCOSERVER

J2EE – Integration Server (XIO)

In the J2EE Administration tool the following changes must be made:

rfcengine

Under *Cluster* → *Server* → *Services* → *rfcengine*
choose the RFC destination *AI_RUNTIME_XIO* and (on the right side) change the entries *Gateway host* and *Application Server* from <server1> to <virtual_1>.

DBPool

Under *Cluster* → *Server* → *Services* → *dbpool*
choose the Available Pools *XIRun* and (on the right side) change the *Database URL* from <server1> to <virtual_1>.

SAPIntegration

Under *Cluster* → *Server* → *Services* → *security*
choose the tab page *SAPIntegration* and change the entry *Application Server* from <server1> to <virtual_1>.

DBConnect Properties

In both XI Systems (XIO, XIS) change the value *hostname* from <server1> to <virtual_1> in the file `/sapmnt/<SID>/global/dbconnect.properties`.

Changes to the Integration Builder

J2EE – Integration Server (XIS)

In the J2EE Administration tool the following changes need to be made:

rfcengine

Under *Cluster* → *Server* → *Services* → *rfcengine*
choose the RFC destination *AI_ DIRECTORY_XIO* and change (on the right side) the entries *Gateway host* and *Application Server* from <server1> to <virtual_1>.

DBConnect Properties

In both XI systems (XIO, XIS) change the value hostname from <server1> to <virtual_1> in the file `/sapmnt/<SID>/global/dbconnect.properties`.

Changes to the Exchange Profile

The Exchange Profile is called at `http://<server2>:53000/ExchangeProfile`. The name must be changed from <server1> to <virtual_1> under

- *Connections* → *com.sap.aii.connect.integrationserver.name*
- *Connections* → *com.sap.aii.connect.rwb.name*

Changes to the System Landscape Directory (SLD)

The SLD is called at `http://<server2>:53000:/SLD`.

Choose *Content Maintenance* and then select *Subset: Landscape Description* and *Class: BC SYSTEM*.

Select a SAP_BCSystem and then choose *Move* and change the occurrence of <server1> to <virtual_1> (in the fields *Name*, *Caption* and *System Home*). Save the changes using the *Create* button. Proceed in the same manner for the following classes:

- BC Client
- BC Application Server
- BC Message Server
- Http Service Port (change here the URL)

Changes to the RFC Adapter

In the properties file for the RFC Adapter change the *hostname* in sections 3 and 4 to the virtual hostname and check the correct system number and user parameters.

# 3. SECTION FOR ACCESS TO XMB INBOUND RFC SERVICE
RfcAdapter.toXmb.host=…
RfcAdapter.toXmb.sysnr=…
# 4. SECTION FOR REGISTRATION OF XMB OUTBOUND RFC SERVICE
RfcAdapter.fromXmb.host=…
RfcAdapter.fromXmb.sysnr=….
SECTION FOR ACCESS TO XMB INBOUND RFC SERVICE

Changes to the Standalone Adapter Engine

In the file `../techn_adapter/BaseConfiguration\SLDaccessor.properties` check the following values:

```
SLD.host=<server2>
SLD.port=53000
SLD.user=XIAPPLUSER
SLD.password=XIPASS
```

There are further changes to be made for property files that are needed for the standalone Adapter Engine. See the scenario descriptions in the next chapter for more details.

Chapter 4: XI Test Scenarios

Overview

This chapter provides configuration instructions for scenarios that can be used to test the implementation of the XI Integration Server in a switchover environment. It is assumed that the XI Integration Server system has been setup in conformance with the previous chapter with switchover addressing implemented. Pre- and post-switchover tests are described.

Contents

Introductory Remarks

Only the major channels into the XI Integration Server are considered in this chapter: XI-to-XI, RFC and IDoc communication. Plain HTTP access should work, if XI-to-XI communication is feasible, because the same underlying ICF technology (Internet Communication Framework) is used here (provided addresses are used correctly for HTTP plain access).

Scenario 1 and 2 focus on asynchronous communication (QualityOfService = ExactlyOnce), while scenario 3 is a synchronous scenario. However, there is basically no difference between synchronous and asynchronous access in XI, because the underlying communication technology is identical. If addressing works with asynchronous connections, synchronous calls will work as well and the other way round.

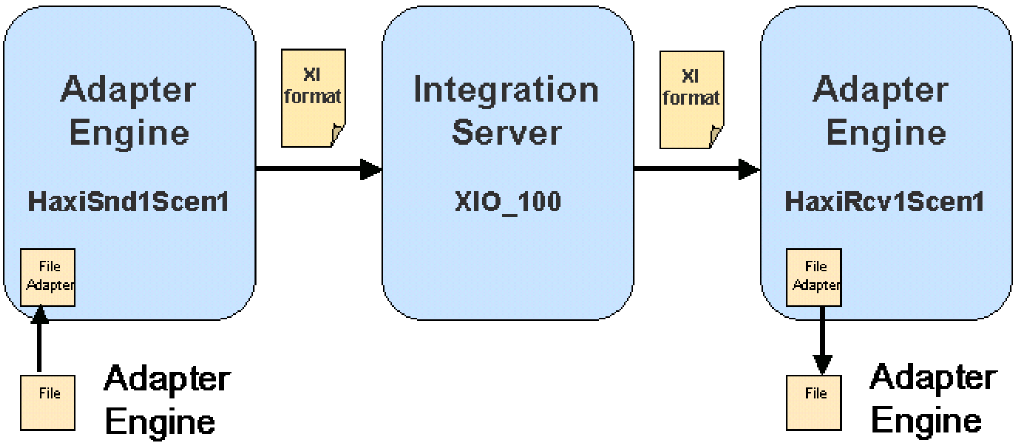
Scenario 1 – Native XI Communication

This section describes a communication scenario where both sender and receiver natively support the XI message format and protocol. Usually this scenario will occur when both application systems have XI support built-in (Integration Engines in SAP systems with SAP WEB AS Release 6.20 or higher). These application systems normally contain applications that use generated XI proxies to directly communicate in XI manner (in other words no adapters are required).

Setting up additional SAP systems for testing issues creates considerable installation and administration effort. That is why this XI-to-XI scenario is simulated here by implementing both sender and receiver systems as simple file adapters (residing in the Adapter Engine).

The sender adapter sends native XI messages to the Integration Server. The receiver adapter receives native XI messages from the Integration Server. From the point of view of the Integration Server there is no difference between file adapters and Integration Engines. Thus for the issue of interest here – testing the XI Integration Server in a switchover environment - this simplified scenario can be used to provide a test for the desired scenario of native XI communication.

The scenario is set up as follows:



Configuring the Adapter Engine

Preparation

Please obtain the Zip files HaxiScen1RfcAdapter.zip and HaxiScen1Repository.zip

you to the correct location on the SAP Marketplace for download.

Install the sample configuration files for the standalone RFC adapter. Unzip the file **HaxiScen1RfcAdapter.zip** in the directory `/usr/sap/adapter`. This ZIP file contains the sample configuration files used for this documentation. Start the standalone adapter after copying the configuration files.

Sender

The sender for the Adapter Engine is called **HaxiSnd1Scen1**. The following configuration tasks have to be performed.

1. Start the Adapter Engine on <server2>
2. Call the Adapter Engine administration at
   `http://<server2>:8300`
3. Under *Adapters* select HaxiSnd1Scen1
4. Choose *Configure* → *Edit/View*
5. Change the Value for *XMB.TargetURL* to
   `http://<virtual_1>:8000/sap/xi/engine/?type=entry`
6. Choose *Store Configuration Data*
7. After this change the adapter has to be started with the *Restart* button.

Receiver

The receiver for the Adapter Engine is called **HaxiRcv1Scen1**. The following configuration tasks have to be performed.

1. Start the Adapter Engine on <server2>
8. Call the Adapter Engine administration at
   `http://<server2>:8300`
2. Under *Adapters* select HaxiRcv1Scen1
3. Choose *Configure*
4. The value for the receiving port is set to XMB.httpPort=7738
   Make sure that this port does not conflict with an existing port.

Configuring the Integration Repository

The entries for the Integration Repository may be imported. Start the Integration Builder (Design).

1. Get the following files from the SAP Service Marketplace:
   HaxiScen1Repository.zip
2. Copy the files to <server2>:
   XI2_0_HAXI_1_of_sap.com.xi_1.tpt
   XI2_0_HAXI_1_of_sap.com.xi_classic.tpt
   to the directory
   `/usr/sap/XIS/SYS/global/xi/repository_server/import`
3. Choose *Object* → *Import XI 2.0 Development Objects*
4. Select XI2_0_HAXI_1_of_sap.com.xi_1
5. In the follow up screen choose *Resolve Collisions*
6. The import will be confirmed

7. Repeat the steps 2 to 5 for XI2_0_HAXI_1_of_sap.com.xi_classic
8. The software component HAXI is available in the Integration Repository

Configuring the Integration Directory

Create an Receiver Endpoint

The receiver will be called **HaxiRcv1Scen1**. To create this receiver call the Integration Builder (Configuration).

1. Choose *Object* → *New* → *End Point*
2. Enter the following values:
   Business System HaxiRcv1Scen1
   Interface Name HaxiRcv1Scen1
   Interface Namespace http://sap.com/xi/haxi
3. In the following form enter the following values
   Endpoint Type XI Connectivity
   Addressing Mode URL
   URL http://<server2>:7738/file/receiver
4. Save and activate the configuration

Interface Determination

1. Choose *Object* → *New* → *Interface Determination*
2. Enter the following values:
   Business System HaxiSnd1Scen1
   Interface Name HaxiSnd1Scen1
   Interface Namespace http://sap.com/xi/haxi
   Receiver HaxiRcv1Scen1
3. In the following form enter the values
   Inbound Interface Name HaxiRcv1Scen1
4. Choose the mapping rule 
   Mapping Rule HaxiScen1
5. Save and activate the configuration

Receiver Determination

1. Choose *Object* → *New* → *Receiver Determination*
2. Enter the following values:
   Business System HaxiSnd1Scen1
   Interface Name HaxiSnd1Scen1
   Interface Namespace http://sap.com/xi/haxi
3. In the following form enter the values
   Receiver HaxiRcv1Scen1
4. Save the configuration
5. Press the *Refresh* button in the configuration overview. No errors should be reported.

Scenario Test

This scenario will send a test message from the sender **HaxiSnd1Scen1** to the receiver **HaxiRcv1Scen1**. In order to test the scenario the file adapter **HaxiSnd1Scen1** has to be started inside the XI Adapter Engine. After startup it will send all XML messages found in the directory ***scen1*** to the Integration Server. The Integration Server will process the message and send it to the receiver **HaxiRcvScen1** that persists the message to a file named xmb_output.xml.

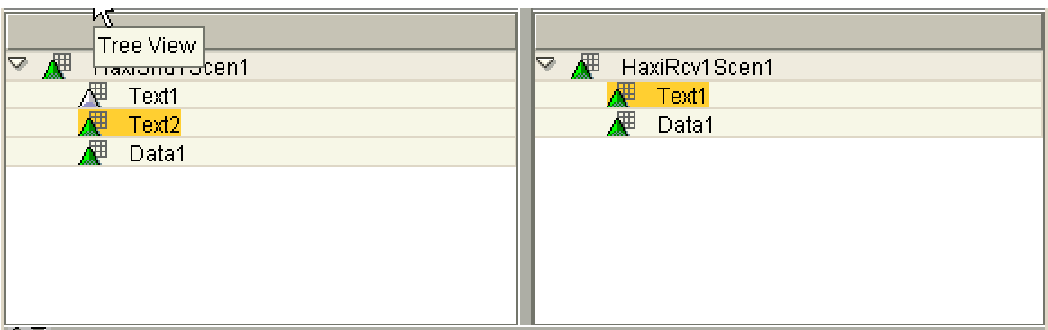
Checking the scenario execution requires checking the existence and contents of the output file on the receiver and checking for errors inside the Integration Server.

For this reason the following structure exists under the top-level directory `/usr/sap/adapter` (UNIX filesystem notation is used here, use Windows notation if appropriate). The directory scen1 will contain the Sample File `test.xml` with the following content:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ns0:HaxiSnd1Scen1 xmlns:ns0="http://sap.com/xi/haxi">
  <Text1>Input</Text1>
  <Text2>Output</Text2>
  <Data1>1</Data1>
  </ns0:HaxiSnd1Scen1>
```

This message is mapped in the Integration Server to the format required by the receiving adapter. The mapping modifications are simple:

- The field Text2 will be mapped to Text1.
- The original content of Text2 will be dropped.



Test Execution

The following actions need to be performed to start and verify the correct function of this scenario:

1. Start the Adapter Engine on <server2>

2. Call the Adapter Engine Administration at `http://<server2>:8300`
3. Under *Adapters* select **HaxiSnd1Scen1**
4. Stop the adapter if it is running and delete the log
5. Restart the adapter. Do not use the *Start* button after configuration changes.
6. View the adapter's log. The log should state that `test.xml` was sent to the Integration Server successfully.
7. Stop the adapter, otherwise the messages would be repeated every 20 seconds.
8. Check if the file `xmb_output.xml` has been created in the *rcv1* Directory. The mapped message should look as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
 <ns0:HaxiRcv1Scen1 xmlns:ns0="http://sap.com/xi/haxi">
  <Text1>Output</Text1>
  <Data1>1</Data1>
  </ns0:HaxiRcv1Scen1>
```

9. Logon to the Integration Server system (XIO, client 100).
10. Call the message monitoring transaction SXMB_MONI in order to verify that the status is OK. The status OK means that the message has been received, processed and sent out to the receiver successfully.

Single Message Test

In order to perform a basic test, only a single file `test.xml` is transmitted. This can be done before and after switchover to prove basic connectivity and functionality for the Integration Server.

1. Send a single file `test.xml` (before a switchover is performed).
2. Check that the message arrived at the receiver and that it was processed to status OK in the Integration Server.
3. Perform the switchover of the Integration Server to the standby cluster node.
4. When the Integration Server is available again after switchover, send a test file again to perform a basic functions test again.
5. Again, check that the message arrived at the receiver and that it was processed to status OK in the Integration Server.

Message Mass Tests

In order to perform a test that is closer to real-world switchover, create a constant sender side workload to the Integration Server across a switchover taking place. Of course some files will fail to be sent as soon as the Integration Server fails. However, the files should be re-sent after switchover and startup again. In addition there should be no hanging messages in the Integration Server (because of the retry mechanisms working in side the Integration Server).

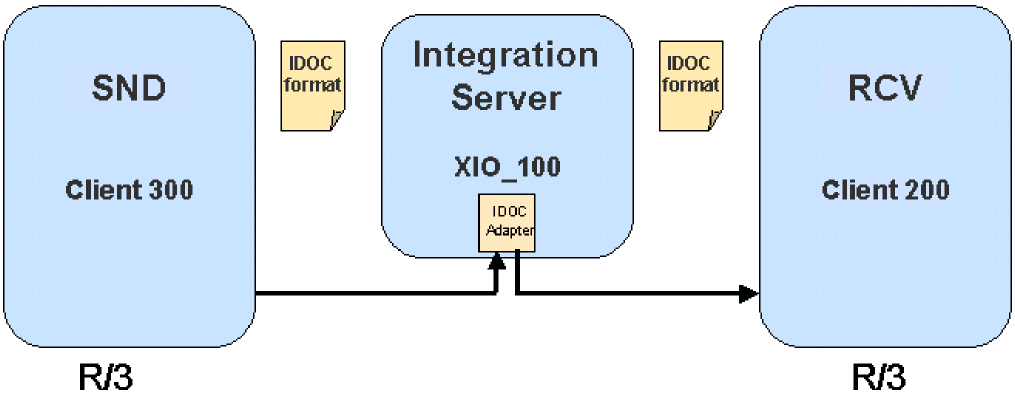
1. Create several copies of the file `test.xml` (in other words copy `test.xml` to `test1.xml`). Since all files in the source directory will be processed by the file adapter, you can increase the workload. This will ensure a continuous workload towards the Integration Server as long as files remain to be processed.
2. Start sending the files (before a switchover is performed - and keep sending them across the switchover happening).
3. Check that the messages do arrive at the receiver and that they are processed to status OK in the Integration Server.
6. Perform the switchover of the Integration Server to the standby cluster node. (The file adapter should be unable to reach the Integration Server now.)
7. When the Integration Server is available again after switchover, check that new messages do arrive at the receiver. Check that all messages have been processed to status OK in the Integration Server. Note that you may have to wait for some time until this is achieved because automatic retries are being performed by the Integration Server runtime (10 minutes waiting time should be enough)

Scenario 2 – IDoc Communication

This section describes a communication scenario based on the native SAP IDoc format for both sender and receiver. This scenario may occur when both application systems do not have XI support built-in (for example SAP Web AS releases prior to 6.20), but IDocs are chosen as the means of communication.

Different clients of the XIS system will be used as the IDoc sender and IDoc receiver systems here. The sender sends a native IDoc to the IDoc adapter residing inside the Integration Server. After processing the IDoc, the server sends out another IDoc through its built-in IDoc adapter to the receiver system.

The scenario is set up as follows:



It is possible to use existing clients in an R/3 system or Web AS system as a sender system (SND) and receiver system(RCV).

It is also possible to use two different clients of one system both as sender and receiver.

This documentation is based on the type of input needed for a SAP Web AS 6.20. If

system number and the client used in this documentation are only substitutes, which need to be to be replaced with the values of your specific sender and receiver system.

Configuring the Sender System

Sender SND Client 300

Create an RFC destination for the connection to the Integration Server residing on XIO Client 100.

1. Logon to SND client 300
2. Call the transaction SM59 (see below)

Create an RFC Connection

1. Tab page *Technical Settings*

RFC Destination	XIO_100
Connection Type	3
Description	XIO Client 100
Load Distribution	Yes
Target System	XIO
Msg. Server	<virtual_1>
Group	Public
Gateway Host	<virtual_1>
Gateway Service	sapgw20
Tab page *Logon/Security*	
Language	English
User	xiappluser
Password	xipass
Tab page *Special Options*	
Unicode	depending on System

2. Test the Connection
   Test → Authorization

Create a Port Definition for IDoc Processing

1. Call transaction WE21
2. Choose *Transactional RFC*
3. Create a new port

Portname	generate port name (example: A000000039)
Description	XIO client 100
RFC_Destination	XIO_100

4. Save the Port

Create a Partner Profile

1. Call transaction WE20
2. Choose *Partner Type LS*
3. Create a new Logical System
4. In *Outbound parameters* add a new line .
5. In the form *Partner Profiles: Outbound Parameters*

Partner No.	XIO_100
Partner Type	LS
Message Type	SYIDOC
Receiver Port	the port definition (which means A000000039)
Pack. Size	1
Output Mode	transfer IDoc immed.
Basic Type	SYIDOC01

6. Save the parameters
7. Save the partner profile

Configuring the Integration Directory

Create an Receiver Endpoint

The receiver will be the Client 200 in the RCV system. To configure this receiver call the Integration Builder (Configuration).

1. Choose *Object* → *New* → *End Point*
2. Enter the following values:

Business System	RCV_200
Interface Name	SYIDOC.SYIDOC01
Interface Namespace	urn:sap-com:documents:sap:idoc:messages

3. In the form enter the following values

Endpoint Type	IDOC-Adapter
RFC Destination	RFC_200
Segmentversion	620 (depending on your target system)
Interface version	4.0
Port	SAPRCV
SAP Release	620 (depending on your target system)

4. Save and activate the configuration

Interface Determination

1. Choose *Object* → *New* → *Interface Determination*
2. Enter the following values:

Business System	SND_300
Interface Name	SYIDOC.SYIDOC01
Interface Namespace	urn:sap-com:documents:sap:idoc:messages
Receiver	RCV_200

3. In the form enter the following values

Inbound Interface Name	RCV_200

5. Save and activate the configuration

Receiver Determination

1. Choose *Object* → *New* → *Receiver Determination*
2. Enter the following values:
   Business System SND_300
   Interface Name SYIDOC.SYIDOC01
   Interface Namespace urn:sap-com:documents:sap:idoc:messages
3. In the form enter the following values
   Receiver RCV_200
4. Save the configuration
5. Press the *Refresh* button in the *Configuration Overview*. No errors should be reported.

Configuring the Integration Server

RFC Destination for Receiver

Create an RFC destination for the connection to the receiver system RCV client 200.

1. Logon to XIO Client 100
2. Call the transaction SM59
3. Create an RFC Connection
   Tab page *Technical Settings*
   RFC Destination RCV_200
   Connection Type 3
   Description RCV Client 200
   Load Distribution Yes
   Target System RCV
   Msg. Server <hostname>
   Group Public
   Gateway Host <hostname>
   Gateway Service sapgw00
   Tab page *Logon/Security*
   Language English
   User xisuper
   Password xipass
   Tab page *Special Options*
   Unicode Enable Unicode
4. Test the connection
   Test → *Authorization*

Port Maintenance in IDoc Adapter

5. Call the transaction IDX1
6. Choose *Create*
   Port SAPSND
   Client 300

Description	SND Client 300
RFC Destination	SND_300

Verify, that the user XISUPER has SAP_ALL permissions in SND client 300. This is

Configuring the Receiver System

Create a Partner Profile

1. Logon to RCV Client 200
2. Call transaction WE20
3. Choose *Partner Type LS*
4. Create a new logical system
5. In *Inbound Parameters* add a new line .
6. In the form *Partner Profiles: Inbound Parameters*

Partner No.	XIO_100
Partner Type	LS
Message Type	SYIDOC
Process Code	APLI
Processing by Function M.	Trigger Immediately

7. Save the parameters
8. Save the partner profile

Scenario Test

Basic Test

In order to test the scenario IDocs have to be sent from the Client 300 in XIS to the Integration Server. For this purpose you have to create an IDoc using the IDoc test tool.

1. Logon to the sender system (SND Client 300)
2. Call the transaction WE19
3. Choose *Basic Type*
4. Enter SYIDOC01 for the basic type
5. Choose *IDOC* → *Create*
6. Choose *IDOC* → *Edit Control Record*

Sender	
Port	SAPSND
Partner Number	SND_300
Partner Type	LS
Receiver	
Port	The port definition (which means A000000039)

Partner Number XIO_100
Partner Type LS

7. Choose *IDOC → Test Outbound IDOC*
8. Number of IDOC 1
9. A success message should be displayed.

Verify Processing

1. Logon to the Integration Server XIO Client 100
2. Call the transaction SXMB_MONI
3. The Status and the Adapter Status should be OK
4. Logon to the Receiver System, RCV Client 200
5. Call the transaction WE02
6. The IDoc should be added. Since it could not be processed you can ignore the red traffic light here.

Failover Testing (Mass Test)

1. Make sure, that enough workload is created in the XI Integration Server. You can do this by sending multiple IDocs in the IDoc testing tool.
2. Perform the switchover of the Integration Server to the standby cluster node.
3. When the Integration Server is available again, check that the messages in retry mode are re-sent finally.
4. Finally all messages should be in the Status OK.

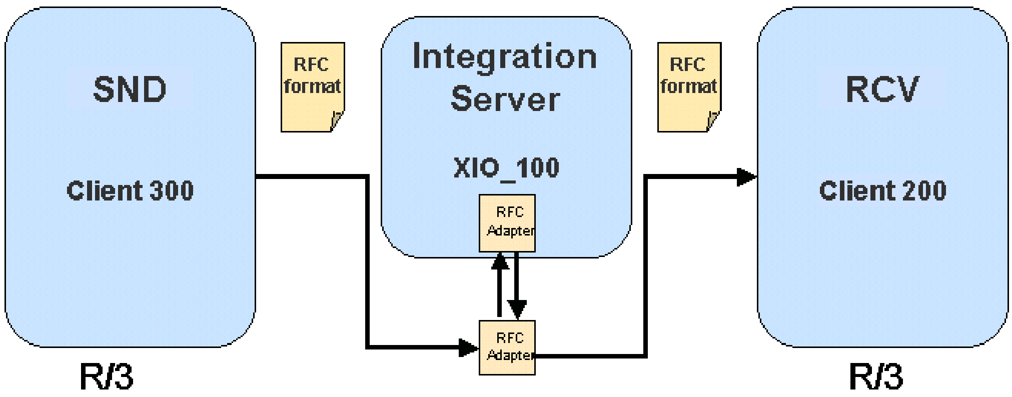
Scenario 3 – RFC Communication

This section describes a communication scenario based on native SAP RFC calls for both sender and receiver. This scenario may occur when both application systems do not have XI support built-in (for example SAP Web AS releases prior to release 6.20), so direct RFC has to be chosen as the means of communication.

Different clients of the XIS system will be used as the RFC sender and RFC receiver systems here. The sender sends a RFC call to the installed RFC adapter. After processing the RFC call, the Integration Server sends out another RFC call through the RFC adapter to the receiver system. This will test the functions of the inbound and outbound RFC adapter and their connections to the Integration Server.

> This communication scenario establishes a synchronous (QualityOfService =
>
> Integration Server. All other test scenarios have worked on as asynchronous models (QualityOfService = *ExactlyOnce* in XI terminology).

The scenario is set up as follows:



It is possible to use existing clients in an R/3 system or Web AS system as a sender system (SND) and receiver system (RCV).

It is also possible to use two different clients of one system both as sender and receiver.

This documentation is based on the typed input needed for a SAP Web AS 6.20. If

system number and the client used in this documentation are only substitutes, which need to be to be replaced with the values of your specific sender and receiver system.

Configuring the Sender System

RFC Destination for Receiver

Create an RFC destination for the connection to the sender system SND client 300.

1. Logon to SND Client 300
2. Call the transaction SM59
3. Create an RFC connection
   Tab page *Technical Settings*

RFC Destination	SND_RFC
Connection Type	T
Description	Link to RFC adapter
Registered Server Program	X
Program ID	SND2RFC

4. Make sure, that the RFC adapter in the Integration Server is configured and started.

5. Test the connection
   Test → *Connection*
6. Create an RFC connection
   Tab page *Technical Settings*

RFC Destination	XIO_100_RFC
Connection Type	L
Description	Sender to Integration Server
Reference Entry	SND_RFC

Configuring the Integration Directory

Create a Receiver Endpoint

The receiver will be the client 200 in the RCV system. To configure this receiver call the Integration Builder (Configuration).

1. Choose *Object* → *New* → *End Point*
2. Enter the following values:

Business System	RCV_200
Interface Name	BAPI_SFLIGHT_GETLIST
Interface Namespace	urn:sap-com:documents:sap:rfc:functions

3. In the form enter the following values

Endpoint Type	RFC-Adapter
Load Balancing Activated	X
Message Server	<hostname>
Service	3600
System ID	RCV
Logon Group	Public

   Save and activate the configuration

Logon Data

1. Choose *Object* → *New* → *Logon Data*

Enter the following values:

Business System	RCV_200
Interface Name	BAPI_SFLIGHT_GETLIST
Interface Namespace	urn:sap-com:documents:sap:rfc:functions

In the form enter the following values

1. Authentication Procedure — Password-Based

Authentication Procedure	Password-Based
SAP Client	200
User Name	USER
Password	XIPASS
Logon Language	English

2. Save and activate the configuration

Interface Determination

1. Choose *Object* → *New* → *Interface Determination*
2. Enter the following values:

Business System	RCV_300
Interface Name	BAPI_SFLIGHT_GETLIST
Interface Namespace	urn:sap-com:documents:sap:rfc:functions

3. In the form enter the following values

Inbound Interface Name	BAPI_SFLIGHT_GETLIST
Namespace	urn:sap-com:documents:sap:rfc:functions

4. Save and activate the configuration

Receiver Determination

1. Choose *Object* → *New* → *Receiver Determination*
2. Enter the following values:

Business Scenario	HAXI
Business System	SND_300
Interface Name	BAPI_SFLIGHT_GETLIST
Interface Namespace	urn:sap-com:documents:sap:rfc:functions

3. In the form enter the following values

Receiver	RCV_200

4. Save the configuration
5. Press the *Refresh* button in the *Configuration Overview*. No errors should be reported.

Configuring the Integration Server

RFC Adapter

Add the following entries to the properties file of the RFC adapter. Change the file `rfc_adapter.properties on <server1>`.

```
#   1.2 connectivity settings
#
RfcAdapter.registerInbound= <keep_existing_entries> sourceBackendXIS
# 2. SECTION FOR REGISTRATION TO CLIENT APPLICATION SYSTEMS
#
RfcAdapter.sourceBackendSND.host=<hostname>
RfcAdapter.sourceBackendSND.sysnr=00
RfcAdapter.sourceBackendSND.progid=SND2RFC
RfcAdapter.sourceBackendSND.poolsize=2
RfcAdapter.sourceBackendSND.unicode=1 (depending on source system)
RfcAdapter.sourceBackendSND.client=300
```

RfcAdapter.sourceBackendSND.passwd=XIPASS

RfcAdapter.sourceBackendSND.lang=DE

RfcAdapter.sourceBackendSND.trace=5

After applying these changes stop and restart the Adapter Engine.

1. Start the Adapter Engine on <virtual_1>
2. Call the Adapter Engine administration at
   http://<virtual_1>:8200
3. Choose *rfc_adapter*
4. Choose *stop*
5. Choose *restart*

Scenario Test

In order to test the scenario, RFC calls will be made from the sender system (via the RFC adapter) to the Integration Server, which will then (via the RFC adapter) call the receiver system (synchronous connection through all systems).

Basic Testing

1. Logon to the sender system (SND Client 300)
2. Call the transaction SE37
3. Enter the function module BAPI_SFLIGHT_GETLIST
4. Choose *Function module* → *Test* → *Single Test*
5. Enter the following values

RFC Target System	XIO_100_RFC
FROMCOUNTRYKEY	DE
FROMCITY	FRANKFURT
TOCOUNTRYKEY	US
TOCITY	NEW YORK
AIRLINECARRIER	LH
AFTERNOON	
MAXREAD	0

6. Choose *Function Modules* → *Execute*
7. The call will be returned, if it has been processed.

Verify that the call was processed in the Integration Server.

1. Logon to the Integration Server (XIO Client 100)
2. Call the transaction IDX5
3. Choose *RFC Monitor*
4. Choose *Program* → *Execute*

5. You should see a INBOUND and a OUTBOUND XML Message for the Function Module BAPI_SFLIGHT_GETLIST

Failover Testing

This scenario uses a synchronous communication, so there are no messages to be monitored in the Integration Server persistent database. The only reasonable testing option is to repeat this basic synchronous call before and after switchover has been performed.

In order to automatically perform a series RFC calls, you may implement a test program (ZTEST) like the following.

Note that the RFC may or may not be called 'in background task'. Normal RFC will lead to a synchronous call with QualityOfService = *BestEffort*, while a call 'in background task' will lead to asynchronous handling with QualityOfService = *ExactlyOnce* automatically. The distinction between both call options is in the caller code. The RFC adapter will correspondingly set the QualityOfService in the XI message.

```
REPORT  ZTEST .

PARAMETERS: DEST LIKE RFCDES-RFCDEST DEFAULT 'XIO_100_RFC',
           ZMAX TYPE I DEFAULT 10.



DATA: ZFLIGHTLIST TYPE TABLE OF BAPISFLIST,
      ZRETURN TYPE BAPIRET2,
      ZCOUNTER TYPE I VALUE 1.

while ZCOUNTER <= ZMAX.

CALL FUNCTION 'BAPI_SFLIGHT_GETLIST'
IN BACKGROUND TASK
DESTINATION DEST
  EXPORTING
     FROMCOUNTRYKEY = 'DE'
     FROMCITY = 'FRANKFURT'
     TOCOUNTRYKEY = 'US'
     TOCITY = 'NEW YORK'
     AIRLINECARRIER = 'LH'
     MAXREAD = 1
  TABLES
```

```
CALL FUNCTION 'START_OF_BACKGROUNDTASK'
     EXPORTING
          STARTDATE = SY-DATUM
          STARTTIME = SY-UZEIT
     EXCEPTIONS
          OTHERS    = 1.

IF SY-SUBRC = 1.
  EXIT.
ENDIF.
COMMIT WORK.

add 1 to zcounter.
endwhile.
```

Chapter 5: Recommendations

Overview

This section gives some recommendations for administrative and check actions that may be required after a switchover has happened in a productive environment.

Contents

Integration Server Issues

Message Restarts

Restart Incorrect Messages

If messages remain in error states after switchover, you should start these messages again manually from the XI monitor or using the report RSXMB_RESTART_MESSAGES (see SAP note 602256; this report is part of Support Package XI 2.0 SP2).

Do not use the transaction SMQR to restart messages, because this will result in a short dump (displayed in ST22). You cannot restart XI messages by using native qRFC (see XI configuration parameters MONITOR, QRFC_RESTART_ALLOWED). See SAP note 608261 for more information.

tRFC

Call transaction SM58 and check through the list. If necessary, start hanging tRFC calls under the *Edit* menu using the command Execute LUW.

You may also use the report RSARFCEX with variants.