

# SAP Exchange Infrastructure Tuning Guide



## SAP XI Release 3.0

Document Version: 1.0

---

## Copyright

© Copyright 2004 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft<sup>®</sup>, WINDOWS<sup>®</sup>, NT<sup>®</sup>, EXCEL<sup>®</sup>, Word<sup>®</sup>, PowerPoint<sup>®</sup> and SQL Server<sup>®</sup> are registered trademarks of Microsoft Corporation.

IBM<sup>®</sup>, DB2<sup>®</sup>, DB2 Universal Database, OS/2<sup>®</sup>, Parallel Sysplex<sup>®</sup>, MVS/ESA, AIX<sup>®</sup>, S/390<sup>®</sup>, AS/400<sup>®</sup>, OS/390<sup>®</sup>, OS/400<sup>®</sup>, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere<sup>®</sup>, Netfinity<sup>®</sup>, Tivoli<sup>®</sup>, Informix and Informix<sup>®</sup> Dynamic Server<sup>™</sup> are trademarks of IBM Corp. in USA and/or other countries.

ORACLE<sup>®</sup> is a registered trademark of ORACLE Corporation.

UNIX<sup>®</sup>, X/Open<sup>®</sup>, OSF/1<sup>®</sup>, and Motif<sup>®</sup> are registered trademarks of the Open Group.

Citrix<sup>®</sup>, the Citrix logo, ICA<sup>®</sup>, Program Neighborhood<sup>®</sup>, MetaFrame<sup>®</sup>, WinFrame<sup>®</sup>, VideoFrame<sup>®</sup>, MultiWin<sup>®</sup> and other Citrix product names referenced herein are trademarks of Citrix Systems, Inc.

HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C<sup>®</sup>, World Wide Web Consortium, Massachusetts Institute of Technology.

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 expressly 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/nw04> -> Documentation -> Operations -> SAP XI

## Content

<b>1</b>	<b>Introduction</b> .....	<b>5</b>
1.1	Target Groups.....	5
1.2	Naming Conventions .....	5
<b>2</b>	<b>Architecture of SAP Web Application Server</b> .....	<b>7</b>
<b>3</b>	<b>Tuning SAP Exchange Infrastructure</b> .....	<b>8</b>
3.1	SAP XI Configuration and Tuning .....	8
3.1.1	SAP Profile Parameters.....	8
3.1.2	XI Configuration Parameters .....	9
3.1.3	Load Balancing .....	10
3.1.3.1	HTTP .....	11
3.1.3.2	RFC Logon .....	11
3.1.3.3	SAP Gateway .....	11
3.1.3.4	qRFC .....	12
3.1.4	J2EE Adapter Framework.....	12
3.1.5	Miscellaneous .....	13
3.1.5.1	Additional qRFC Features .....	13
3.1.5.2	IDoc Adapter .....	13
3.1.5.3	BPM.....	14
3.1.5.4	Archiving .....	14
3.1.5.5	Switch Deletion .....	14
3.2	Java VM Tuning.....	15
3.3	Database Tuning .....	15
3.4	Application Tuning .....	15
3.4.1	Message Size .....	15
3.4.2	XI Mappings.....	16
<b>4</b>	<b>Performance Analysis</b> .....	<b>17</b>
4.1	Performance Measurement .....	17
4.2	Statistical Records .....	17
4.3	SAP Services.....	17
<b>5</b>	<b>Appendix</b> .....	<b>18</b>
5.1	References .....	18

# 1 Introduction

This document covers the use of SAP Exchange Infrastructure (SAP XI) Release 3.0 on SAP Web Application Server (Web AS) 6.40 in a mission-critical or high-volume environment.

The basic concepts and tuning parameters for SAP XI and Web AS 6.40 are discussed. It also provides best practice information on configuration and the relevant settings for tuning. Experiences are based on performance tests and several internal and external projects.

## 1.1 Target Groups

This document is aimed at anyone wanting to run SAP XI in a mission-critical environment and who therefore needs both background information and advice on how to set up and run SAP XI to ensure optimal performance.

Note that you should be familiar with the concepts and details of SAP Web AS. It is also necessary to be familiar with the general concepts of SAP XI and its runtime environment.

This documentation is aimed at the following groups:

- Consultants
- Customers using SAP XI 3.0 (System Administrators)

## 1.2 Naming Conventions

### Terms and Abbreviations

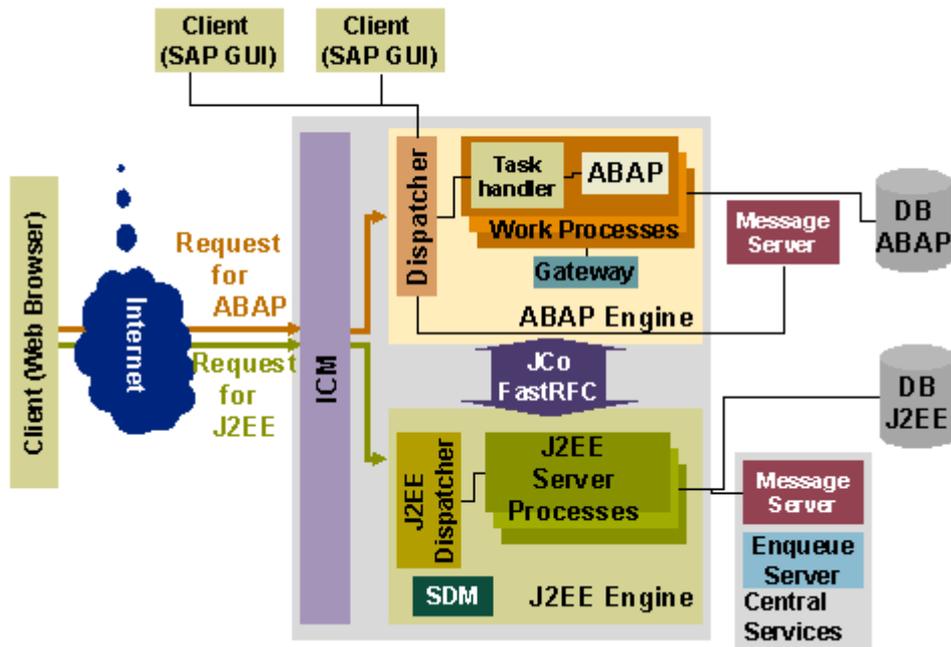
The following terms and abbreviations will be used in this document:

<b>Term</b>	<b>Explanation</b>
SAP XI	SAP Exchange Infrastructure
Integration Server (IS)	The server where SAP XI processing (routing, mapping, queuing, conversions) takes place. The Integration Server contains a centrally configured Integration Engine and all other required SAP XI components.
Integration Engine (IE)	The SAP XI instance available on an SAP application system with Release 6.20 or higher. The Integration Engine sends messages to, or receives messages from, the central Integration Server.
Quality of Service	SAP XI term for describing how the transmission and processing of messages is to be handled. 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.
ABAP Engine	ABAP part of the SAP Web AS
J2EE Engine	J2EE part of the SAP Web AS
AS	Application server
SID	The standard SAP 3-character system identification
JCo / RFC Provider	Java Connector used for interconnecting JAVA software and SAP systems using RFC.
Gateway	Standard SAP software used for interconnections using RFC and CPIC (often also called CPIC-Gateway).
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 ABAP part of the SAP Web AS 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 File System
A2A	Application-to-application communication typically connects application software within a company (on the intranet).
B2B	Business-to-business communication typically connects application software of one company to the (different) application software of another company (over the Internet).
SLD	System Landscape Directory
ICM	Internet Communication Manager (icman)
GC	Garbage Collection (Java VM)
PMI	Process Monitoring Infrastructure
AF	XI Adapter Framework
DMZ	Demilitarized Zone
SAPS	SAP Application Performance Standard (SAPS) is a hardware-independent unit that describes the performance of a system configuration in the SAP environment.
SMP	Symmetric multiprocessing

## 2 Architecture of SAP Web Application Server

The SAP Web Application Server is the application platform of SAP NetWeaver, which is the basis for the other SAP NetWeaver components. SAP XI 3.0 uses the integrated version with ABAP and Java. Each integrated instance of the SAP Web AS contains the ABAP dispatcher and its work processes, which can process the ABAP programs, as well as the Java dispatcher and its server processes, to which it distributes the incoming J2EE requests.



Within XI, HTTP requests are handled either by ABAP work processes (for example, IS pipeline entry URL) or by J2EE server nodes (for example, Adapter Framework). Communication between ABAP and J2EE engines uses RFC (JCo) or HTTP interfaces. An example of RFC-based communication is the XI mapping runtime, which is called by the XI Integration Server during pipeline processing.

An additional infrastructure component (for example, SAP Web Dispatcher) is installed for HTTP load balancing between several Web AS instances.

## 3 Tuning SAP Exchange Infrastructure

Tuning for SAP Exchange Infrastructure needs to cover the different areas of the application's software stack as well as Java VM, databases, and operating system.

This document focuses mainly on the following tuning areas:

- SAP XI configuration and tuning
- Java VM tuning
- Database tuning
- Application tuning

### 3.1 SAP XI Configuration and Tuning

The configuration of XI Web AS instances is crucial for XI performance and throughput.

Important key factors include:

- SAP profile parameters for basic Web AS system setup
- XI configuration parameters to influence processing characteristics
- Load balancing to distribute and optimize resource consumption

#### 3.1.1 SAP Profile Parameters

Web AS instances should be configured for mass use of parallel RFC (SAP Note 74141) and for high interface load (SAP Note 384971). You can also configure RFC quotas dynamically using ABAP report *RSARFCLD*.

The following table shows the relevant parameters and recommended values:

Parameter	Value	Description
<a href="#">abap/arfcstate_col_delete</a>	X	Activates deletion of <i>ARFCRSTATE</i> records in background. Forces report <i>RSTRFCEU</i> to run in batch periodically every 5 minutes (see SAP Note 539917)
<a href="#">gw/max_conn</a>	2000	Sets maximum number of active connections (Gateway)
<a href="#">gw/max_overflow_size</a>	10000000	Sets size of local memory area for Gateway
<a href="#">icm/HTTP/max_request_size_KB</a>	2097152	Maximum size of HTTP request accepted by ICM
<a href="#">rdisp/appc_ca_blk_no</a>	2000	Sets tcp/ip communication buffer size
<a href="#">rdisp/force_sched_after_commit</a>	no	Disables automatic rollout of context after commit work
<a href="#">rdisp/max_arq</a>	2000	Maximum number of internal asynchronous messages
<a href="#">rdisp/max_comm_entries</a>	2000	Sets maximum number of communication entries
<a href="#">rdisp/rfc_max_own_login</a>	90	Sets RFC quota for own logins
<a href="#">rdisp/rfc_max_own_used_wp</a>	90	Sets RFC quota for own used work processes

<a href="#">rdisp/rfc_max_wait_time</a>	5	Sets RFC maximum wait time after load check
<a href="#">rdisp/tm_max_no</a>	2000	Sets maximum number of available connections (instance)
<a href="#">rdisp/wp_ca_blk_no</a>	2000	Sets block buffer size for work process communication
<a href="#">ztta/max_memreq_MB</a>	2048	Limit for a single memory allocation request (default 64). This value depends on the maximum document size.

As qRFC requests are processed by dialog work processes, you may have to adapt profile parameter `rdisp/max_wprun_time` to prevent `TIME_OUT` failures when processing very large documents.

You configure the number of dialog work processes for each Web AS instance with SAP profile parameter `rdisp/wp_no_dia`. HTTP requests as well as qRFC and tRFC requests are processed by dialog work processes. You set the configured number of processes depending on the available CPUs for processing. Set the following as the initial configuration:

```
rdisp/wp_no_dia = 4 * #CPU
```

### 3.1.2 XI Configuration Parameters

You can set the configuration parameters for the Integration Server by using transaction `SXMB_ADM`.

The following table shows the relevant parameters and recommended values:

Category	Parameter	Value	Description
IDOC	<a href="#">SYNTAX_CHECK</a>	0	You can switch off the IDoc syntax check for well-tested productive scenarios.
RUNTIME	<a href="#">LOGGING</a>	0	Runtime logging persists an XI message after each pipeline processing step.
RUNTIME	<a href="#">TRACE_LEVEL</a>	1	The default setting for the trace level should be 1; only raise this level in error situations.
TUNING	<a href="#">EO_INBOUND_PARALLEL</a>	See below	Global number of parallel XI inbound queues; a parameter for sender-specific queues also exists.
TUNING	<a href="#">EO_INBOUND_TO_OUTBOUND</a>	1	Activates XI outbound queues for EO processing (without message split); can be switched off in specific configurations.
TUNING	<a href="#">EO_OUTBOUND_PARALLEL</a>	See below	Number of parallel XI outbound queues for each receiver
TUNING	<a href="#">EO_MSG_SIZE_LIMIT</a>	20480	Can be set to enable processing of large messages (in this case above 20 MB) in specific inbound queues (XBTL*) so that the processing of average-sized messages is not affected; limit is scenario-specific.

The number of XI processing queues depends on the following:

- Maximum number of requests to be processed in parallel
- Ability and performance of receiver applications (for outbound queues)

As a starting point, set the sum of active (running in parallel) processing queues to  $2 * \#CPU$  (available on all Web AS instances running XI). For example, if you have 2 servers with 4 CPUs each, set the number of parallel queues (inbound and outbound) to 16.

When calculating this value, keep in mind that outbound queues might be blocked for a long time during the `CALL_ADAPTER` step, thus needing only small or no local CPU resources. In this case, these outbound queues should be partly or completely ignored.

Queue parallelism is used to control resource (CPU) consumption on the Integration Server. To optimize resource usage, configure the least number of queues that allows CPU utilization at near 100%. Using more parallel queues will only overload the system and use even more resources. The qRFC scheduler algorithm works most efficiently with fewer queues and more entries for each queue.

Depending on the scenario, you can deactivate XI outbound queue processing in the following cases:

- Only IDoc adapter outbound processing is used (request is processed by tRFC layer)
- Receivers are fast and will not be overloaded
- Expensive content-based routing is not used
- No expensive mappings are processed
- You generally only have a few large messages to process

XI outbound queues are used automatically for XI message splits.

The benefit of deactivating outbound queue processing is lower CPU consumption (20-30%) because there is one less asynchronous step (the persistence step including qRFC processing). However, in most cases we recommend using outbound queue processing as it provides more flexibility.

### 3.1.3 Load Balancing

Load balancing is a key feature for ensuring SMP scalability. XI makes use of load balancing by using multiple Web AS instances. In XI, load balancing is especially useful in inbound request processing. You can use outbound load balancing if the receiver application or component is prepared for and can benefit from load balancing (for example, a receiving IDoc or RFC application is addressed using its message server to distribute the load).

The following table shows areas where XI can use load balancing for inbound processing:

Protocol	Component	Load Balancer	Description
HTTP	Integration Server	HTTP Load Balancer	IS pipeline entry processing using HTTP load balancing.
HTTP	Adapter Framework	HTTP Load Balancer	Adapter Framework inbound processing using HTTP load balancing.
RFC	IDoc Adapter	Message Server	IDoc Adapter inbound processing using logon load balancing.
RFC	RFC Adapter	SAP Gateway	RFC sender channel can register multiple processing threads at a client system's gateway.
RFC	Mapping Runtime	SAP Gateway	JCo RFC Provider service registers multiple threads at IS gateway for receiving mapping requests; these JCO server threads then call an application thread for mapping execution.
RFC	qRFC	qRFC scheduler	qRFC scheduler distributes load across available Web AS instances (can be controlled by RFC server groups).

Different components can be used as the HTTP load balancer (hardware or software), for example:

- SAP Web Dispatcher
- Hardware Load Balancer
- Web Switch

### 3.1.3.1 HTTP

The number of ICM processes (and therefore Web AS instances) is important for HTTP request processing. Besides processing HTTP inbound requests and forwarding them to their respective handler work process, ICM processes also process HTTP client outbound requests (that is, work processes sending HTTP requests using plain HTTP or SOAP outbound adapters).

If you are using several Web AS instances, there are three possible solutions for load balancing of inbound HTTP traffic (without using an external component such as HW Load Balancer):

#### *Message server*

Every client uses the message server HTTP port for addressing purposes (parameter *ms/http\_port*). The disadvantage with this solution is that the message server sends an HTTP redirect to the client, which then resends the request to the specified server, thus causing additional network load.

#### *Web Dispatcher*

A Web Dispatcher can forward HTTP requests. It also gets current load information from the message server to identify instances with lower load. This causes no additional network load and is the preferred solution if using several Web AS instances. Logon groups can be used to control the distribution.

#### *Manual distribution*

In this case, load distribution is performed by client applications, which send requests directly to different ICM processes. For example, several J2SE File adapters are used in parallel, each sending to one specific ICM process.

### 3.1.3.2 RFC Logon

Scenarios using the RFC protocol (RFC Adapter, IDoc) benefit from RFC logon load balancing, which is a standard Web AS function. RFC requests are forwarded by the Web AS message server, which collects information about Web AS instance load. Logon groups can be used to control the distribution.

### 3.1.3.3 SAP Gateway

A server application can register several times at a Web AS gateway with the same program ID. (The program ID is configured in transaction *SM59 RFC Destinations - TCP/IP Connections*). The gateway process then uses request counters and a round-robin algorithm to distribute requests to registered server programs (this is called load distribution by multiple gateway registration).

#### **Mapping Runtime**

Each rfcengine thread of a J2EE server registers at a specific Web AS gateway. The mapping service is called by a work process using the RFC destination *AI\_RUNTIME\_JCOSEVER*. This ensures load distribution between rfcengine threads. It is important to note that each J2EE server registers the bulk of its rfcengine threads at service startup. This may lead to a suboptimal distribution if using several J2EE server nodes.

The gateway process then dispatches the first mapping requests to server node 0, the next to server node 1. The gateway process is not aware of the J2EE configuration. It only dispatches requests to registered server applications. If you encounter such a situation, you can distribute parallel load to several J2EE servers by:

- Lowering the number of rfcengine threads per server node for destination *AI\_RUNTIME\_\** (depending on the number of parallel mapping requests). You configure the number of rfcengine threads (number of processes) for each RFC destination within the JCo RFC Provider service of a J2EE server (using Visual Administrator).

- Using the local gateway. Using local gateways for the mapping runtime instead of one central gateway ensures that more J2EE servers process mapping requests in parallel and prevents overload of the central gateway. To enable the use of an instance's local gateway, just delete the gateway option of the RFC destination *AI\_RUNTIME\_JCOSERVER* in *SM59*. Mapping requests are then distributed to local J2EE server nodes using the local gateway. Therefore, there must be a configured J2EE server node up and running with every Web AS instance.

## RFC Adapter

The RFC adapter can also make use of the multiple server registration feature. This means that an RFC sender channel within the Adapter Framework can register several JCo server threads at a client systems' gateway.

### 3.1.3.4 qRFC

The qRFC scheduler implements load distribution by RFC server groups and by managing available work process resources (see *SMQR* -> *Goto* -> *QRFC Resources*). For a short overview of scheduler configuration and functions, see SAP Note 369007.

## 3.1.4 J2EE Adapter Framework

### XI Message Persistence

Asynchronous messages are persisted during Adapter Framework inbound or outbound processing. You must keep in mind that the XI message table *XI\_AF\_MSG* will grow according to the message throughput. This needs to be observed to prevent table overflows, especially in databases using storage parameters (Oracle, DB2). To estimate how much storage is required, multiply the maximum number of messages to be kept in the database by the average message size and add 20% as a safety margin.

### Database Load Distribution

The central Adapter Framework in an All-In-One installation (Add-In installation with all XI components) uses the same database instance as the Integration Server. In a high-volume processing scenario, one option is to use a separate non-central Adapter Framework instance with its own database and other resources (to be installed on a different host), thus reducing load in the central database of the XI Integration Server.

### File Adapter

You configure File adapter sender channels by specifying one directory where files are located. To guarantee Exactly-Once processing of files, multiple J2EE server nodes are synchronized using enqueue locks when accessing this directory. Therefore, adding J2EE server nodes will not improve the performance of such a scenario. To get a higher degree of parallelism and increase throughput, you can split the files to be processed into different directories. You can then set up several File adapter sender channels to process these files in parallel.

### J2EE Configuration

The Adapter Framework requires a set of fixed application threads within the J2EE Engine for processing different internal messaging queues (AFW queue, BC Adapter, CIDX Adapter, and so on). Additionally, certain channel configurations also require a fixed set of application threads (for example, RFC sender channel, File adapter channels). This may require a higher number of application threads. Therefore, within the J2EE Engine configuration (configtool or Visual Admin), go to:

*Global Configuration* -> *Server* -> *Kernel* -> *ApplicationThreadManager*

Set the *MaxThreadCount* property to 300

## 3.1.5 Miscellaneous

### 3.1.5.1 Additional qRFC Features

#### Prioritization

Queue prioritization for XI processing queues uses a built-in feature of qRFC. By specifying the maximum processing time for a queue, you can define how much processing time a specific queue will get (this works like a time-scheduling procedure, time slicing). You can use this feature to add queues for high-priority and low-priority messages (for inbound and outbound processing). These queues are then named, for example *XBT1\** or *XBT9\** for inbound EO processing and *XBTA\**, *XBTZ\** for outbound EO processing.

This function uses the message filter concept and can be configured in transaction *SXMB\_ADM*. You can define message filters for specific message attributes such as sender/receiver service, interface, and so on. A filter can also be dependent on the message size.

This function can also be useful in situations where a queue backlog needs to be processed, due to a system error or an error on the receiver side, for example. In this case, you can use a prioritized queue for urgent messages that must be processed immediately, without having to process the whole backlog first.

#### Balancing

Balancing EO queues makes it possible to balance the fill level of EO processing queues. This is a useful feature if queue parallelism is set at a higher level (for example, queues are added by raising the value for the *EO\_INBOUND\_PARALLEL* parameter). In general, this is useful if a system has not yet reached the processing facility (and is not near to 100% CPU consumption). If a queue has more entries than the balancing limit, then top-of-queue messages are distributed to queues with a lower fill level.

You can set balancing parameters *BALANCING*, *B\_EO\_IN\_PARALLEL*, *B\_EO\_OUT\_PARALLEL* in XI configuration in transaction *SXMB\_ADM* (category *TUNING*). For more information, see the parameter documentation. You have to switch the *BALANCING* parameter on and off manually.

#### Delayed Processing

You can use message filter criteria to delay message processing. This is especially useful for moving processing of specific messages to a later point in time, when system load is lower than during normal office hours, for example. During EO inbound processing, these messages are not inserted into queues for immediate pipeline processing. EOIO messages are put into their respective processing queue, but the queue is locked for later processing. A batch job then reschedules EO messages to normal processing queues or unlocks the corresponding queue for EOIO messages.

### 3.1.5.2 IDoc Adapter

#### IDoc Packages

IDocs are normally sent in packages between systems. You configure this within the IDoc configuration partner profile by specifying processing mode *Collect IDocs and transfer*. IDocs are then processed by report *RSEOUT00* in the sender system. During inbound processing in the IDoc adapter of the Integration Server, this IDoc package is split up into single XI messages. Therefore, an IDoc package cannot be processed as one unit within XI. This also has an impact on performance.

In SP11, a solution will be available for IDoc adapter outbound processing to collect IDoc-XML messages and transfer them as an IDoc package, as in the traditional IDoc world.

The IDoc adapter is designed to be able to process IDoc-XML structures that represent IDoc packages. This enables IDoc-XML packages to be generated during mapping if the inbound data structure

of a message already includes the package data. This structure can then be processed by the outbound IDoc adapter and sent as an IDoc package to the receiver application.

### **IDoc Acknowledgements**

The IDoc adapter requests acknowledgements (ACKs) by default when processing inbound requests. This leads to additional resource consumption due to ACK handling. For information on how to disable these acknowledgement requests, see SAP Note 777175.

### **qOUT Scheduler for tRFC**

The IDoc outbound adapter transfers outbound requests to the tRFC layer on the Integration Server. You can control the processing of tRFC requests by using the qOUT scheduler, which is normally used to process qRFC outbound requests. You can use transaction *SMQS* to set the number of parallel connections for a receiver RFC destination, or even stop scheduling.

### **3.1.5.3 BPM**

Performance behavior of BPM scenarios is determined by the actual process model and the number of executed process instances of a process.

The following are the determining factors for a particular process:

- Number of messages received by a process (influenced by use of correlations)
- Number of messages sent by a process
- Number of transformations within a process (influenced by processed message sizes)

In general, process steps that generate work items internally are relevant for process performance. Transformation steps (either for message split or merge) also include a mapping step and can be quite expensive. This should be taken into consideration during scenario design.

For an example of resource consumption, see the BPM scenario described in the XI sizing guide:

*The inbound SOAP messages are handed over to the BPE, which triggers a two-step process consisting of message reception and sending. The processed messages are then sent to a receiver by using the SOAP channel; no mapping.*

The SAPS value for this scenario, in which 10 kB messages are processed, is 1300 (processing one process instance each second). A simple XI proxy scenario, for example, requires 140 SAPS processing one message each second. This gives you an idea of the resource requirements using BPM.

### **3.1.5.4 Archiving**

A parallel archiving infrastructure is available for archiving XI messages within the Integration Server. You make the configuration settings by using transaction *SXMB\_ADM*. Each archiving process then gets a particular chunk of messages to archive and write to its own archiving file. This allows parallel processing and reduces overall archiving time.

### **3.1.5.5 Switch Deletion**

The reason behind the switch deletion procedure is the fact that deleting records in databases can become quite an expensive task. This procedure switches to a second table container and copies the necessary XI message data into this container, then drops the whole table. Before data can be archived, it is important that XI messages have an archivable status (check the adapter status and the message state).

The switch deletion procedure has the following advantages:

- If you need to delete a large number of messages from the database, INSERT data + DROP table is faster than DELETE. Of course, the amount of data to be copied has a major impact on the performance of this procedure; therefore, the retention time of XI messages should be small.
- Since tables (and indexes) are dropped often, this procedure reduces overall table fragmentation and space requirements (especially if table or index fragmentation is an issue).

## 3.2 Java VM Tuning

Java VM tuning is one of the most crucial tuning steps, especially for more complex scenarios. For information about setting baseline JVM parameters, see SAP Note 723909. You must also take platform-specific parameters into account (for example, JIT compiler settings). The impact of Garbage Collection (GC) behavior especially may become a critical issue. Overall GC times for the J2EE application should be well below 5%. For more information about GC behavior and settings, see also SAP Note 552522.

Specific to XI is the fact that you sometimes need to process large documents for mapping or when using signatures. This can lead to excessive memory usage on the Java side. Therefore, you must observe Garbage Collection and the available Java heap in order to evaluate performance and prevent OutOfMemory exceptions. Since XI mapping is processed by stateless session beans that are called using a JCo interface, this may lead to a reduction of parallel JCo server threads within the JCo RFC Provider service of a J2EE server node (you can compensate for this by adding J2EE server nodes).

## 3.3 Database Tuning

The database is of crucial importance for overall system performance since XI persists the messages it transports several times.

### I/O Distribution

Distributing I/O evenly across available I/O channels (controllers, disks) is a key factor for optimal database configuration. Therefore, you should use as many independent channels as possible. Separate data areas from log areas (as well as from paging/swap areas).

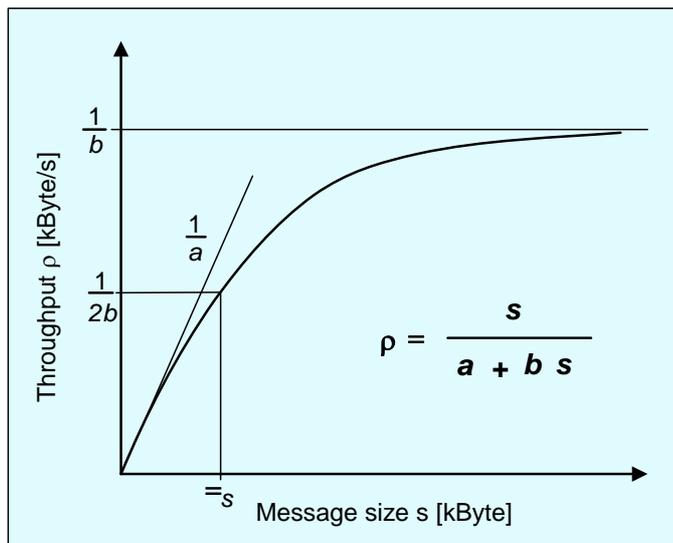
The XI application profile is characterized by the following:

- High-volume processing in a small number of tables (about 10)
- Some tables (qRFC) with high insert/delete frequency

## 3.4 Application Tuning

### 3.4.1 Message Size

Throughput in terms of bytes per second is much higher for larger messages. This is mainly due to the processing overhead for a single message (see figure). Assuming a linear model for the cost of processing a single message of size  $s$ :  $\tau = a + bs$ , the throughput  $\rho$  increases linearly with message size for small messages and asymptotically reaches a maximum for large messages.



Furthermore, the Integration Engine stores messages in a compressed format (compression rates for XML messages are normally in the range of 70-80%). Therefore, applications should be designed to use larger messages (if possible). On the other hand, processing larger messages causes higher memory resource consumption. Therefore, you should consider 64bit platforms as there are no address space issues to limit memory usage.

### 3.4.2 XI Mappings

XI basically provides two types of mappings:

- XSLT mappings (processed by the SAP XSLT processor)
- Java mappings (XI Mapping Toolkit or customer development)

Resource consumption for the mapping depends on the complexity of the mappings and the source document sizes. In general, XSLT mappings require more memory than mapping classes generated by the XI Mapping Toolkit.

Since mappings are processed by the J2EE Engine, the maximum available Java heap may be a limiting factor for the maximum document size the XI mapping service is able to process. Tests have shown that processing of XSLT mappings consumes up to 20 times the source document size (using identity mapping). The maximum available Java heap for 32bit JVMs is platform-dependent. Using 64bit JVM platforms is an option here.

Current maximum heap sizes – 32bit

OS	Maximum heap (GB)
Linux	2
Windows	1.2 – 1.4

The Java heap is limited by the heap limit of the process (may be limited by address space because operating system code or libraries may also be loaded within the same address space). Also, Java internal memory areas such as the permanent space for loading Java classes must fit into the same address space.

## 4 Performance Analysis

### 4.1 Performance Measurement

The XI Integration Server provides performance measurement based on time stamps. You can activate this in transaction *SXMB\_ADM* by setting the *MEASUREMENT\_LEVEL* parameter in category *PERF*. A value greater than zero causes the Integration Engine to write a time stamp in the message at the beginning and end of each processing step. These time stamps appear in the *Performance-Header* header attribute of an XI message. Additionally, the *MEASUREMENT\_PERSIST* parameter causes the timestamp data to be persisted in a separate table for evaluation.

To view the data measured, you can use the Performance Monitor in the XI Runtime Workbench. There, you can display data at a detailed time-stamp level or in an aggregated form. For information about running specific aggregation jobs to consolidate data, see the documentation.

You also have the option of using statistics delivered by PMI (Process Monitoring Infrastructure) to evaluate cross-component performance data. This enables you to include the J2EE Adapter Framework, for example, in measurement data. For information on how to set up PMI and the corresponding data collection jobs, see the documentation.

### 4.2 Statistical Records

You can use statistical records for performance evaluation. XI processing normally generates the following statistical record types:

- HTTP (for HTTP inbound processing)
- RFC (for asynchronous calls during qRFC processing)

Use transactions ST03N, STAD to analyze statistical records.

For example, an XI message processed in an EO scenario using IENGINE SOAP-HTTP for inbound and outbound processing (without using an outbound queue) generates the following statistical records output:

Task Type	#Steps	Average Request Time	Average CPU Time	Average DB Time
HTTP	1	72	63	9
RFC	3	114,7	57	16

This means that there is one HTTP record for inbound processing (in this example running the respective ICF handler class) and 3 RFC records for asynchronous processing steps. These RFC records are triggered by qRFC processing and include XI processing (pipeline, outbound adapter).

XI you can also use statistical records to determine memory usage. For example, when using SOAP-HTTP for inbound and outbound processing, the maximum memory consumption per message is:

3 MB + 5 \* XML document size

This formula does not consider any additional features such as content-based routing or mapping.

### 4.3 SAP Services

For a description of the services provided by SAP Service&Support for performance analysis, see SAP Note 812158.

## 5 Appendix

### 5.1 References

[Sun2003] Performance Documentation for the Java HotSpot Virtual Machine.  
Sun Microsystems.

URL: <http://java.sun.com/docs/hotspot/>