Interstage
Shunsaku Data Manager
8.0

Manual Update List

November 2006

J2UZ-8970-01ENZ0(A)

Preface

▼ Purpose of This Manual

This manual describes information added, deleted, or modified in the manuals for Interstage Shunsaku Data Manager Enterprise Edition 8.0 (hereafter referred to as Shunsaku).

▼ Organization of This Manual

This manual only describes the Shunsaku manuals that have been updated, and separates them into chapters as in the following table:

Chapter	Updated Shunsaku Manual	Manual code
Chapter1	Interstage Shunsaku Data Manager Operator's Guide	J2X1-1941-01ENZ2(03)
Chapter2	Interstage Shunsaku Data Manager Application Development Guide	J2X1-1152-02ENZ2(01)
Chapter3	Interstage Shunsaku Data Manager Command Reference	J2X1-1911-01ENZ2(02)
Chapter4	Interstage Shunsaku Data Manager Troubleshooting	J2X1-3650-02ENZ2(01)

▼ Notations

Platform-specific Information

All Shunsaku manuals describe information relevant to all platforms supported by Shunsaku. Where the description differs between platforms, the applicable platform is indicated for the respective descriptions.

Refer to the information applicable to the platform in use.

Windows

Describes details relevant to Windows

Linux

Describes details relevant to Linux

Solaris

Describes details relevant to Solaris(TM) Operating System

Abbreviations

Abbreviations	Product name
Shunsaku	Interstage Shunsaku Data Manager Enterprise Edition for Windows, Interstage Shunsaku Data Manager Enterprise Edition for Linux, and Solaris Interstage Shunsaku Data Manager Enterprise Edition
Windows	Microsoft(R) Windows(R) 2000 Server operating system, Microsoft(R) Windows(R) 2000 Advanced Server operating system, Microsoft(R) Windows Server(TM) 2003, Standard Edition, Microsoft(R) Windows Server(TM) 2003, Enterprise Edition, Microsoft(R) Windows Server(TM) 2003, Standard x64 Edition, and Microsoft(R) Windows Server(TM) 2003, Enterprise x64 Edition
Windows 2000	Microsoft(R) Windows(R) 2000 Server operating system, and Microsoft(R) Windows(R) 2000 Advanced Server operating system
Windows Server 2003	Microsoft(R) Windows Server(TM) 2003, Standard Edition, Microsoft(R) Windows Server(TM) 2003, Enterprise Edition, Microsoft(R) Windows Server(TM) 2003, Standard x64 Edition, and Microsoft(R) Windows Server(TM) 2003, Enterprise x64 Edition
Solaris	Solaris(TM)9 Operating System, and Solaris(TM)10 Operating System
Linux	Red Hat Enterprise Linux AS (v.3 for x86), Red Hat Enterprise Linux ES (v.3 for x86), Red Hat Enterprise Linux AS (v.4 for x86), and Red Hat Enterprise Linux ES (v.4 for x86)
Symfoware Server	Symfoware Server Enterprise Extended Edition, and Symfoware Server Enterprise Edition
Studio	Interstage Studio Standard-J Edition 8.0, and Interstage Studio Enterprise Edition 8.0
Interstage Application Server	Interstage Application Server Web-J Edition 7.0, Interstage Application Server Plus 7.0, Interstage Application Server Standard Edition 7.0, Interstage Application Server Enterprise Edition 7.0, Interstage Application Server Standard-J Edition 8.0, and Interstage Application Server Enterprise Edition 8.0
Application Server Plus Developer	Interstage Application Server Plus Developer V7.0
Frameworks Suite	Interstage Application Frameworks Suite Web Edition V7.0, Interstage Application Frameworks Suite Standard Edition V7.0, and Interstage Application Frameworks Suite ClientJ Edition V7.0

Symbols/Marks/Icons

Symbols/Marks/Icons	Meaning
Shaded text	Indicates added, changed, or deleted text.
:	Indicates text that has been deleted. Example: <Sentence> : : <Sentence>
⚠ Note	Indicates an important note.

▼ How to Read This Manual

This manual explains each chapter that has been updated in each Shunsaku manual. The update list is on the first page of each chapter. Confirm where and when (month, year, edition number) the updates are made then refer to the details of the updates.

November/2006

Update history
November/2006, First edition

Microsoft, Visual C++, Windows, and Windows Server are registered trademarks or trademarks of Microsoft Corporation U.S.A. in the United States and other countries.

Solaris and all other trademarks and logos related to Solaris are registered trademarks or trademarks of Sun Microsystems Inc. U.S.A. in the United States and other countries, and are used under license from the said company.

Linux is a registered trademark or trademark of Linus Torvalds in the United States and other countries.

Java and other trademarks comprising Java are trademarks of Sun Microsystems Inc. U.S.A. in the United States and other countries.

Red Hat, RPM, and all trademarks and logos based on Red Hat are registered trademarks or trademarks of Red Hat Inc. in the United States and other countries.

Oracle is a registered trademark of Oracle Corporation.

Interstage is registered trademark of Fujitsu Limited.

Table of Contents

Chapter 1 Interstage Shunsaku Data Manager Operator's Guide

This chapter explains the updates in Interstage Shunsaku Data Manager Operator's Guide (J2X1-1941-01ENZ2(03)).

Update List

Number	Updated section	Description	Month/year updated (Edition number)
	Overall	Corrected misprints	August/2006 (First Edition)
	Overall	Corrected misprints	October/2006 (Second Edition)
1.1	2.2.3 Designing the Data Structure	Polished explanations	November/2006 (Third Edition)
1.2	2.2.5 Estimating Resource Requirements	Polished explanations	November/2006 (Third Edition)
1.3	B.2.2 Setting the MaxConnection Parameter and the MultiMax Parameter	Polished explanations and corrected symbols	November/2006 (Third Edition)

1.1 Updates in “2.2.3 Designing the Data Structure”

Description	Polished explanations
Month/year updated (Edition number)	November/2006 (Third Edition)

1.1.1 Previous Version

The data structure of XML documents stored in Shunsaku must be designed by taking into account a unit of data handled in an application.

The processing for XML documents stored in Shunsaku system is processed in a unit of XML document. The number of hits from a search by specifying a search expression is the number of XML documents that is hit. In addition, the sort and aggregation process specified by a return expression and a sort expression are also processed in XML document unit. The update process is also updated in XML document unit.

As a result, units of an access to Shunsaku system are united to an XML document unit.

Same element name can be repeated in an XML document.

However, because Shunsaku processes in XML document unit, it is necessary to design the system by considering which XML document unit is to be used in an application, then include those elements into 1 XML document.

For example, if specify a search expression to perform the search in multiple XML document, the number of hits will be counted to 1 even more than one elements in an XML document are hit.

For the data below, the data in the left is effective for the usage of data classified by factories, however, the data in the right is appropriated for the usage of data classified by employees.

1.1.2 Current Version

The data structure of XML documents stored in Shunsaku must be designed by taking into account a unit of data handled in an application.

The processing for XML documents stored in Shunsaku system is processed in a unit of XML document. The number of hits from a search by specifying a search expression is the number of XML documents that is hit. In addition, the sort and aggregation process specified by a return expression and a sort expression are also processed in XML document unit. The update process is also updated in XML document unit.

As a result, units of an access to Shunsaku system are united to an XML document unit.

Same element name can be repeated in an XML document.

However, because Shunsaku processes in XML document unit, it is necessary to consider which XML document unit is to be used in an application and design the system so that each XML document has those elements.

For example, if specify a search expression to perform the search in multiple XML document, the number of hits will be counted to 1 even more than one elements in an XML document are hit.

For the data below, the data in the left is effective for the usage of data classified by factories, however, the data in the right is appropriated for the usage of data classified by employees.

1.2 Updates in “2.2.5 Estimating Resource Requirements”

Description	Polished explanations
Month/year updated (Edition number)	November/2006 (Third Edition)

1.2.1 Previous Version

Estimating server resources involves taking into account both file resources and memory resources. Refer to Appendix D Estimating Resources for more information on how to estimate each type of resource.

Setting the maximum value of resource and each Shunsaku process can prevent an insufficiency of memory or disk resource used in Shunsaku. By preventing resource from a shortage, the stable operation of Shunsaku system is guaranteed.

Set the maximum value of each resource or each process to each environment file.

Refer to Appendix A Execution Parameters in Environment Files for more information on the environment files.

1.2.2 Current Version

Estimating server resources involves taking into account both file resources and memory resources. Refer to Appendix D Estimating Resources for more information on how to estimate each type of resource.

Setting the maximum value of resource and each process handled in Shunsaku can prevent an insufficiency of memory and disk resource used in Shunsaku. By preventing resource from a shortage, the stable operation of Shunsaku system is guaranteed.

Set the maximum value of each resource or each process to each environment file.

Refer to Appendix A Execution Parameters in Environment Files for more information on the environment files.

1.3 Updates in "B.2.2 Setting the MaxConnection Parameter and the MultiMax Parameter"

Description	Polished explanations and corrected symbols
Month/year updated (Edition number)	November/2006 (Third Edition)

1.3.1 Previous Version

Parameter name	Consideration point
MultiMax (Director environment file)	Specifies the number of multiple requests from search and sort or aggregation process that can be processed at once. If the director receives search requests more than the value specified in this parameter, the exceeded search requests have to be queued and waited until the next process. It is necessary to consider the number of access at a peak time for the value of this parameter. The status where the director executes the search process at the value specified in MultiMax parameter (maximum number of executions that can be executed at once) and the conductor receives requests more than the parameter value, is the status where requests cannot be processed in a peak time. In this case, It is necessary to improve the search performance or to control the number of access by reducing the value of MaxConnection parameter. It is necessary to take into account the condition where normal search requests and requests for sort or aggregation are occurred at the same time, and calculate the value of (normal search requests + requests for sort or aggregation) as the maximum value of this parameter.

Parameter name	Consideration point
MultiMax (Sorter environment file)	Specifies the maximum number of sort or aggregation processes that can be processed at once when process the sort or aggregation in parallel. **One Sorter Configuration** For one sorter configuration, the maximum number of records that the sort and aggregation processes can be processed at once in Shunsaku system is the value of the MultiMax parameter in a sorter. If conductor receives sort or aggregation requests more than the value of MultiMax parameter in sorter, the error is notified to the source invoking the function. The target data have to be searched when process the sort and aggregation. Therefore, the value of MultiMax parameter in director including the execution numbers of search process for sort and aggregation must be set. As a result, the value of both MultiMax parameter must satisfy the following conditions: ● The value of MultiMax parameter in director * the value of MultiMax parameter in sorer To avoid the error of the exceeding maximum value of executions at once in sort process, the following condition must be satisfied: ● The value of MaxConnection parameter * the value of MultiMax parameter in sorter **Multiple Sorters Configuration** For multiple sorters configuration, the maximum number of records that the sort and aggregation processes can be processed at once in Shunsaku system is the total value of MultiMax parameter in all sorters. If conductor executes a sort process at the value of MultiMax in all sorters (the number of sort and aggregation processes that are executing is equal to the total number of MultiMax parameter in all sorters), the conductor does not execute the new receiving process, and the error is notified to the source invoking the function. Because the value of MultiMax parameter in director includes the search process to identify data for sorting and aggregating, it must be set more than the total value of MultiMax parameter in all sorters. Therefore, the value of both MultiMax parameter must satisfy the following conditions: ● The value of MultiMax parameter in director * the total value of MultiMax parameter in all sorters To avoid the error of the exceeding maximum value of executions at once in sort process, the following condition must be satisfied: ● The value of MaxConnection parameter * the total value of MultiMax parameter in all sorters

Parameter name	Consideration point
MaxConnection	This parameter must be set more than the value of MultiMax parameter in director and the total value of MultiMax parameter in all sorters. This parameter is the maximum value of requests that Shunsaku can receive. It is not the number of process that director can execute at once. This means that when [the value of MaxConnection parameter > the value of MultiMax parameter], the conductor receives search requests and issues the search to the director regardless of whether or not the director is processing the search. If the director is not executing the search process, the director will execute the search process received from the conductor using the value specified at the MultiMax parameter in director as the maximum number of processes to be executed at once. For other conditions, the search requests will be in waiting status. : :

The relationship of the value of MaxConnection parameter in the conductor environment file and the value of MultiMax parameter in the director and sorter environment files are as follows:

- MaxConnection * MultiMax (Director environment file)
  The request that exceed the value of director MultiMax parameter is not queued and the error is notified to the invoking source.

- MaxConnection * MultiMax (Director environment file) x 2
  If concurrent processes are assumed to be executed, it is necessary to consider on the peak time accesses.
  The value of the MaxConnection parameter is considered as the maximum number of requests that conductor receives when director is executing the search process (including the search process for sort or aggregation) with the maximum number of executions that can be executed at once (the value of MultiMax parameter).
  However, maximum value of MaxConnection parameter must not be set more than the double value of MultiMax parameter in director (the value of MultiMax parameter in director x 2).

1.3.2 Current Version

Parameter name	Consideration point
MultiMax (Director environment file)	Specifies the number of multiple requests from search and sort or aggregation process that can be processed at once. If the director receives search requests more than the value specified in this parameter, the exceeded search requests have to be queued and waited until the next process.

	It is necessary to consider the number of access at a peak time for the value of this parameter. When the number of requests the conductor receives is more than the parameter value while the director is executing the search process at the value specified in the MultiMax parameter (maximum number of executions that can be executed at once), this means that requests during the peak time cannot be processed. In this case, It is necessary to improve the search performance or to control the number of access by reducing the value of MaxConnection parameter. It is necessary to take into account the condition where normal search requests and requests for sort or aggregation are occurred at the same time, and calculate the value of (normal search requests + requests for sort or aggregation) as the maximum value of this parameter.

MultiMax (Sorter environment file)	Specifies the maximum number of sort or aggregation processes that can be processed at once when process the sort or aggregation in parallel. **One Sorter Configuration** For one sorter configuration, the maximum number of records that the sort and aggregation processes can be processed at once in Shunsaku system is the value of the MultiMax parameter in a sorter. If conductor receives sort or aggregation requests more than the value of MultiMax parameter in sorter, the error is notified to the source invoking the function. The target data have to be searched when process the sort and aggregation. Therefore, the value of MultiMax parameter in director including the execution numbers of search process for sort and aggregation must be set. As a result, the value of both MultiMax parameter must satisfy the following conditions: ● The value of MultiMax parameter in director ≧ the value of MultiMax parameter in sorer To avoid the error of the exceeding maximum value of executions at once in sort process, the following condition must be satisfied: ● The value of MaxConnection parameter ≦ the value of MultiMax parameter in sorter **Multiple Sorters Configuration** For multiple sorters configuration, the maximum number of records that the sort and aggregation processes can be processed at once in Shunsaku system is the total value of MultiMax parameter in all sorters. If conductor executes a sort process at the value of MultiMax in all sorters (the number of sort and aggregation processes that are executing is equal to the total number of MultiMax parameter in all sorters), the conductor does not execute the new receiving process, and the error is notified to the source invoking the function. Because the value of MultiMax parameter in director includes the search process to identify data for sorting and aggregating, it must be set more than the total value of MultiMax parameter in all sorters. Therefore, the value of both MultiMax parameter must satisfy the following conditions: ● The value of MultiMax parameter in director ≧ the total value of MultiMax parameter in all sorters To avoid the error of the exceeding maximum value of executions at once in sort process, the following condition must be satisfied: The value of MaxConnection parameter ≦ the total value of MultiMax parameter in all sorters
:	:

MaxConnection	This parameter must be set more than the value of MultiMax parameter in director and the total value of MultiMax parameter in all sorters. This parameter is the maximum value of requests that Shunsaku can receive. It is not the number of process that director can execute at once. This means that the conductor receives search requests and issues the search to the director regardless of whether or not the director is processing the search during [the value of MaxConnection parameter > the value of MultiMax parameter]. If the director is not executing the search process, the director will execute the search process received from the conductor using the value specified at the MultiMax parameter in director as the maximum number of processes to be executed at once. For other conditions, the search requests will be in waiting status. : :

The relationship of the value of MaxConnection parameter in the conductor environment file and the value of MultiMax parameter in the director and sorter environment files are as follows:

● MaxConnection ≦ MultiMax (Director environment file)
The request that exceed the value of director MultiMax parameter is not queued and the error is notified to the invoking source.

MaxConnection ≦ MultiMax (Director environment file) x 2
If concurrent processes are assumed to be executed, it is necessary to consider on the peak time accesses.
The value of the MaxConnection parameter is considered as the maximum number of requests that conductor receives when director is executing the search process (including the search process for sort or aggregation) with the maximum number of executions that can be executed at once (the value of MultiMax parameter).
However, maximum value of MaxConnection parameter must not be set more than the double value of MultiMax parameter in director (the value of MultiMax parameter in director x 2).

Chapter 2 Interstage Shunsaku Data Manager Application Development Guide

This chapter explains the updates in Interstage Shunsaku Data Manager Application Development Guide (J2X1-1152-02ENZ2(01)).

Update List

Number	Updated section	Description	Month/year updated (Edition number)
2.1	6.3 Format of the Direct Access Key	Polished explanations	November/2006 (First Edition)

2.1 Updates in “6.3 Format of the Direct Access Key”

Description	Polished explanations
Month/year updated (Edition number)	November/2006 (First Edition)

2.1.1 Previous Version

Specify the same element nodes existed in an XML document to the direct access key. If an entity reference is included in the direct access key element and a symbol represented is specified in the direct access key by the entity reference, the return result may not be returned correctly.

2.1.2 Current Version

Specify the same element nodes existed in an XML document to the direct access key. If an entity reference is included in the direct access key element and a symbol represented by the entity reference is specified in the direct access key, the return result may not be returned correctly.

Chapter 3 Interstage Shunsaku Data Manager Command Reference

This chapter explains the updates in Interstage Shunsaku Data Manager Command Reference (J2X1-1911-01ENZ2(02)).

Update List

Number	Updated section	Description	Month/year updated (Edition number)
	Overall	Corrected misprints	October/2006 (First Edition)
	Overall	Corrected misprints	November/2006 (Second Edition)

Chapter 4 Interstage Shunsaku Data Manager Troubleshooting

This chapter explains the updates in Interstage Shunsaku Data Manager Troubleshooting (J2X1-3650-02ENZ2(01)).

Update List

Number	Updated section	Description	Month/year updated (Edition number)
4.1	3.2.2 Log Files	Corrected file formats	November/2006 (First Edition)
4.2	3.2.3 Performance Log Files	Corrected file formats	November/2006 (First Edition)
4.3	3.5.2 Director Data File	Corrected misprints	November/2006 (First Edition)
4.4	A.1.4 Conflict Relationships between Applications and Commands	Polished explanations	November/2006 (First Edition)
4.5	A.2.1 Servers/Network	Polished explanations	November/2006 (First Edition)
4.6	A.2.2 Format of Data	Polished explanations	November/2006 (First Edition)
4.7	A.6.4 Changing Server Configuration	Polished explanations	November/2006 (First Edition)

4.1 Updates in “3.2.2 Log Files”

Description	Corrected file formats
Month/year updated (Edition number)	November/2006 (First Edition)

4.1.1 Previous Version

[Format of the File]

- If the LogFileSwitch parameter is not specified
  - *process identifier*.log
  - *process identifier_generation number*_old.log (collect only if exists)

Example: For the director (*process identifier :* shund1)

```
shund1.log
shund1_1_old.log
```

- If the LogFileSwitch parameter is specified
  - *process identifier*_0_YYYYMMDDhhmm.log
  - *identifier_generation number* _old_YYYYMMDDhhmm.log (collect only if exists)

Example: For the director (*process identifier :* shund1)

```
shund1_0_200510220000.log
shund1_1_old_200510211235.log
```

Remark

process identifier: each process identifier
generation number: the number that the generation is displayed
YYYY: year, MM: month, DD: day, hh: hour, mm: minute

4.1.2 Current Version

[Format of the File]

- If the LogFileSwitch parameter is not specified
  - *process identifier*.log
  - *process identifier_generation number*_old.log (collect only if exists)

Example: For the director (*process identifier :* shund1)

```
shund1.log
```

```
shund1_1_old.log
```

- If the LogFileSwitch parameter is specified
  - *process identifier*_0_YYYYMMDDhhmm.log
  - *process identifier_generation number* _old_YYYYMMDDhhmm.log (collect only if exists)

Example: For the director (*process identifier :* shund1)

```
shund1_0_200510220000.log
shund1_1_old_200510211235.log
```

Remark

process identifier: each process identifier
generation number: the number that the generation is displayed
YYYY: year, MM: month, DD: day, hh: hour, mm: minute

4.2 Updates in “3.2.3 Performance Log Files”

Description	Corrected file formats
Month/year updated (Edition number)	November/2006 (First Edition)

4.2.1 Previous Version

Collecting Performance Log Files

Collect all the files that exist in the path specified at the PfmFileFolder parameter in the environment file of each process.

[Storage Destination: PfmFileFolder parameter (Default)]

Solaris Linux

```
PfmFileFolder    /var/opt/FJSVshnsk/log/ Process Name /
```

Windows

```
PfmFileFolder    "C:\Program Files\Interstage Shunsaku\Shu
nsaku\log\ Process Name \"
```

Remark

Process Name: "conductor", "director", "sorter" or "searcher"

[Format of the File]

- If the PfmFileSwitch parameter is not specified
  - *process identifier*_pfm.log
  - *process identifier*_pfm_*identifier*_*generation number* _old.log (collect only if exists)

Example: For the director (*process identifier :* shund1)

```
shund1_pfm.log
shund1_pfm_1_old.log
```

- If the PfmFileSwitch parameter is specified
  - *process identifier*_pfm_0_YYYYMMDDhhmm.log
  - *process identifier*_pfm_*identifier*_*generation number* _old_YYYYMMDDhhmm.log
    (collect only if exists)

Example: For the director (*process identifier :* shund1)

```
shund1_pfm_0_200510220000.log
shund1_pfm_1_old_200510211235.log
```

Remark

process identifier: each process identifier
generation number: the number that the generation is displayed
YYYY: year, MM: month, DD: day, hh: hour, mm: minute

4.2.2 Current Version

Collecting Performance Log Files

Collect all the files that exist in the path specified at the PfmFileFolder parameter in the environment file of each process.

[Storage Destination: PfmFileFolder parameter (Default)]

Solaris Linux

```
PfmFileFolder     /var/opt/FJSVshnsk/log/ Process Name /
```

Windows

```
PfmFileFolder     "C:\Program Files\Interstage
Shunsaku\Shunsaku\log\ Process Name \"
```

Remark

Process Name: "conductor", "director", "sorter" or "searcher"

[Format of the File]

- If the PfmFileSwitch parameter is not specified
  - process identifier_pfm.log
  - process identifier_pfm_generation number _old.log (collect only if exists)

Example: For the director (process identifier : shund1)

```
shund1_pfm.log
shund1_pfm_1_old.log
```

- If the PfmFileSwitch parameter is specified
  - process identifier_pfm_0_YYYYMMDDhhmm.log
  - process identifier_pfm_generation number _old_YYYYMMDDhhmm.log (collect only if exists)

Example: For the director (process identifier : shund1)

```
shund1_pfm_0_200510220000.log
shund1_pfm_1_old_200510211235.log
```

Remark

process identifier: each process identifier
generation number: the number that the generation is displayed
YYYY: year, MM: month, DD: day, hh: hour, mm: minute

4.3 Updates in “3.5.2 Director Data File”

Description	Corrected misprints
Month/year updated (Edition number)	November/2006 (First Edition)

4.3.1 Previous Version

A director data file is an XML document after imported to Shunsaku.

This is not the requisite information. However, this might be requested when the reproduction test is necessary during the investigation.

The director data file exists at the path specified at the DataFilefolder in the director environment file.

If the director data file cannot be collected according to the customer's convenience, collect the information of the director data file (the information displayed by Is command or dir command).

4.3.2 Current Version

A director data file is an XML document after imported to Shunsaku.

This is not the requisite information. However, this might be requested when the reproduction test is necessary during the investigation.

The director data file exists at the path specified at the DataFilefolder in the director environment file.

If the director data file cannot be collected according to the customer's convenience, collect the information of the director data file (the information displayed by ls command or dir command).

4.4 Updates in “A.1.4 Conflict Relationships between Applications and Commands”

Description	Polished explanations
Month/year updated (Edition number)	November/2006 (First Edition)

4.4.1 Previous Version

However, if the data has been updated or deleted, the data may not be obtained while returning to the client, and the search results may not correct.

4.4.2 Current Version

However, if the data is to be updated or deleted, the search result may not be able to be obtained correctly because there may be no data when obtaining data for returning to the client.

4.5 Updates in “A.2.1 Servers/Network”

Description	Polished explanations
Month/year updated (Edition number)	November/2006 (First Edition)

4.5.1 Previous Version

What has to be considered when operating multiple search servers that are different in hardware specification?

If the machines with different specification, such as the number of CPU block on a search server or the size of memory etc., are used in search process, it is necessary to wait for the lowest performance (slowest) of the search server before returning the search result to an application when executing the search.
As a result, the performance of the whole system depends on the lowest performance search server. Therefore, the servers with same specification are recommended.

Refer to *the Operator's Guide* for more information.

Is there any influence on the Shunsaku operation if director servers and search servers are constructed on the LAN with different segment?

The director servers and search servers in Shunsaku operation can be constructed on the LAN with different segment. However, the LAN environment has to be setup (such as routing) to enable the TCP/IP network between both segments.

For the best performance of Shunsaku, the private line should be used between director servers and search servers. If the data that is not related with the Shunsaku process flows between director servers and search servers, there might be an influence on performance from the network load.

4.5.2 Current Version

What has to be considered when operating multiple search servers that are different in hardware specification?

If the machines with different specification, such as the number of CPU block on a search server or the size of memory etc., are used in search process, the system waits for the search result from the search server whose performance is the lowest (slowest) and then returns the result to an application.
As a result, the performance of the whole system depends on the lowest performance search server. Therefore, the servers with same specification are recommended.

Refer to *the Operator's Guide* for more information.

Is there any influence on the Shunsaku operation if director servers and search servers are constructed on the LAN with different segment?

It is necessary to set the LAN environment (such as routing) so that the TCP/IP network is available between both segments. However, the director servers and search servers in Shunsaku operation can be constructed on the LAN of different segments.

For the best performance of Shunsaku, the private line should be used between director servers and search servers. If the data that is not related with the Shunsaku process flows between director servers and search servers, there might be an influence on performance from the network load.

4.6 Updates in “A.2.2 Format of Data”

Description	Polished explanations
Month/year updated (Edition number)	November/2006 (First Edition)

4.6.1 Previous Version

What format of data can be managed in Shunsaku?

XML documents stored in Shunsaku must be in well-formed format.

However, the preface sections such as XML declarations and DTDs cannot be searched.
In addition, the encoding of XML declarations also cannot be used when storing the data.

Refer to Notes on XML Documents in the *Application Development Guide* for more information.

Multiple XML documents can be imported directly to Shunsaku. This can be done in a single operation if all XML documents are stored sequentially in a single file. Shunsaku manages multiple XML documents in a single file with each separated as one record (data from the start root tag to the end root tag).

4.6.2 Current Version

What format of data can be managed in Shunsaku?

XML documents stored in Shunsaku must be in well-formed format.

However, the preface sections such as XML declarations and DTDs cannot be searched.
In addition, the encoding of XML declarations also cannot be used when storing the data.

Refer to Notes on XML Documents in the *Application Development Guide* for more information.

Multiple XML documents can be imported directly to Shunsaku. This can be done in a single operation if all XML documents are stored sequentially in a single file. Shunsaku manages multiple XML documents in a single file by considering the data between the start root tag and the end root tag as one record.

4.7 Updates in “A.6.4 Changing Server Configuration”

Description	Polished explanations
Month/year updated (Edition number)	November/2006 (First Edition)

4.7.1 Previous Version

Note

Confirm and re-estimate the server resource such as the amount of memory use when adding directors and searchers into the same server.

4.7.2 Current Version

Note

Re-estimate the server resource such as the amount of memory use and confirm that server resource is sufficient when adding directors and searchers in the same server.