IBM WebSphere Partner Gateway 6.2

Tuning Guide

Table of Contents

1 Introduction

This guide describes the application tuning for the various components of IBM WebSphere Partner Gateway 6.2. The objective of this guide is to describe the various tuning parameters for WebSphere Partner Gateway 6.2, including guidance on how to set the values for these parameters.

1.1 About WebSphere Partner Gateway v6.2

WebSphere Partner Gateway v6.2 is IBM's B2B offering designed for exchange of business documents with trading partners. WebSphere Partner Gateway supports a diverse range of business protocols, transport protocols, and security requirements.

Some examples of supported business protocols are AS2, RosettaNet, cXML, ebMS, and so on. WebSphere Partner Gateway 6.2 also supports different transport protocols like HTTP and HTTPS, JMS, SMTP, FTP, SFTP, and so on.

The product allows integration of information. For example, the product can be integrated with a variety of backend systems like IBM WebSphere Process Server, IBM WebSphere Message Broker, or any such backend systems.

With WebSphere Partner Gateway v6.2, all WebSphere Partner Gateway components (except the Integrated FTP Server) are supported on WebSphere Application Server or WebSphere Application Server Network Deployment, which facilitates better administration of WebSphere Partner Gateway. Also, WebSphere Partner Gateway v6.2 makes use of WebSphere Platform Messaging (WPM) for inter-component integration.

WebSphere Partner Gateway v6.2 supports different installation modes, which are briefly explained here.

- Simple Mode

  In Simple Mode, all WebSphere Partner Gateway components are installed on a single WebSphere Application Server and on a single machine. This mode is meant to be used in low volume, demonstration or POC kind of scenarios.

- Simple Distributed Mode

  In Simple Distributed Mode, the WebSphere Partner Gateway components are installed on a single WebSphere Application Server. However, multiple machines are supported within a WebSphere Application Server Network Deployment cell. There is an additional server called Messaging Server created specifically for messaging support. The Messaging Server can either be installed on the same machine that hosts the WebSphere Partner Gateway components or on a separate machine altogether.

- Full Distributed Mode

  In Full Distributed Mode, the WebSphere Partner Gateway components are deployed on separate WebSphere Application Servers. The Messaging Server is also a separate application server. Each component can be installed on multiple machines. The multiple machines are all part of a single WebSphere Application Server Network Deployment cell.

1.2 Purpose

The purpose of this guide is to educate you on the various parameters of WebSphere Partner Gateway 6.2 components, which can be tuned to achieve a better performing system. However, the intent of this guide is not to recommend the values that were used while measuring the performance in the lab environment.

When you apply the recommendations of this guide, it is important to know the baseline performance and the different performance and System utilization metrics. This will help you to identify the performance parameters that require changes. Once this is identified, change the parameters and check the performance of the system to determine the effectiveness of the change.

2 Product tuning aspects

Tuning WebSphere Partner Gateway involves tuning and configuring the different components of the product. This section will describe some common tuning concepts and also some very specific product tuning concepts that can be used to achieve a better performing system.

These tuning concepts involve the following components:

- Application and Messaging Databases
- JVM
- Network
- WebSphere Application Server
- File System
- WebSphere Partner Gateway specific

2.1 Application database tuning

In a typical WebSphere Partner Gateway deployment, the application database (BCGAPPS) is a resource intensive component. The database is used to store configuration information such as trading partner configuration, channel configuration, business capabilities configuration, security information, and so on. During actual processing, the document processing activities are also logged into the database.

It is very important that you place the application database on a separate machine that has high processing capability. The database server should have at least the same number of CPUs as BCGDOCMGR, and the recommended number of CPUs is at least 1.5 times of BCGDOCMGR.

2.1.1 Place database table space and log space on separate disks

It is a common performance strategy to place database log files on physical disks that is different from that of table space containers. It makes a big difference in performance when you physically separate log files from table space containers because it reduces disk contention.

2.1.2 Place database logs on fast disks

Having high performance for read and write access to database log files is critical to the overall WebSphere Partner Gateway system performance.
It is recommended to place database log files on the fastest disks available, with write cache enabled.

2.1.3 Set proper buffer pool size

A buffer pool is an area of memory into which database pages are read, modified, and stored during processing. That is, it is a “cache” used by the database manager to boost the performance, a remedy for slow disk I/O activities.

Generally, having large buffer pools improves performance, but only up to certain limit. Additional memory beyond that limit will not improve the performance. Ensure that the entire buffer pool is accommodated into the real memory, thereby eliminating paging activity on the system. Hence, it is important to actually monitor the status of buffer pool usage of your database by taking DB2 buffer pool snapshots.

By default, WebSphere Partner Gateway has two buffer pools: the default “IBMDEFAULTBP” of 4KB page size, and a custom pool “BUFF32K” of 32KB page size. With a value of 1000 pages for both buffer pools, a very high hit rate (> 99%) for both the pools is observed.

2.1.4 DB2 specific tuning

Though the purpose of this guide is not to provide comprehensive DB2 tuning details, there are a few general rules that can help to improve the performance of DB2.

The following section discusses how to apply these rules. Here are few helpful references for DB2 performance tuning:

1. DB2 UDB Enterprise Edition V8.1: Basic Performance Tuning Guidelines
2. IBM DB2 9 on AIX 5L with NFS, iSCSI, and FCP using IBM System Storage N series
3. DB2 II: Performance Monitoring, Tuning and Capacity Planning Guide
4. DB2 UDB V8 and WebSphere V5 Performance Tuning and Operations Guide
5. DB2 UDB/WebSphere Performance Tuning Guide

You need to have an in-depth understanding to tune DB2 system, and it becomes more challenging with so many available parameters. The first, probably the safest, step in tuning DB2 is to run “DB2 Configuration Advisor” (against your typical work load). From this point, you can always further tune your databases, if required, based on the actual monitoring result.

In order to determine the set of parameters to suggest, the “DB2 Configuration Advisor” requires information about your typical workload against the database. The “DB2 Configuration Advisor” can be started from the DB2 Control Center.
The following parameters can be used to run “DB2 Configuration Advisor”:

- Server memory: **100%**

- Workload type: **Transactions (order entry)**
- Avg. SQL statements per UOW: **More than 10 (longer transactions)**
- Transactions per minutes: **6000**
- Optimize for: **Faster transaction performance**
- Is the database populated with data? **Yes**
- Avg. local apps: **0**
- Avg. remote apps: **200**
- Isolation level: **Cursor stability**

To achieve high performance, it is also equally critical to maintain proper statistics of database tables and indexes. DB2 v8.2 and later introduced automatic statistics collection, also known as auto-runstats. DB2 will automatically perform RUNSTATS in the background to ensure the most current database statistics. From DB2 v9 onwards, features such as automatic configuration, RUNSTATS, self-tuning memory, and automatic storage are enabled by default when you create new databases.

By default, the statistics collected by DB2 auto-runstats are basic table statistics with distribution information, and detailed index statistics using sampling (RUNSTATS options WITH DISTRIBUTION AND SAMPLED DETAILED INDEXES ALL). Also, only tables with high levels of activity (measured through the number of updates, deletes and inserts) are considered by DB2 auto-runstats.

In addition to DB2 auto-runstats utility, you can perform other operations against the database to achieve possibly even better performance. The REORGCHK and REORG are the two commands to enhance DB2 performance. The REORGCHK command checks table and index statistics to determine if any table or index needs to be reorganized or cleaned up. The REORG command actually reorganizes an index or a table. As both commands generate high load on the database system, it is recommended to run these commands only periodically, say once a week, and only during system off-peak period.

The reorgchk command.
db2 –v reorgchk update statistics on table all

It is observed that the following command is required for achieving good performance. It is a good practice to run the reorgchk command before a db2rbind.

```
db2 connect to bcgapps
db2rbind BCGAPPS -l /tmp/rbind.log all –u db2inst1 –p password
```

2.1.5 JDBC data source settings

The following JDBC data sources are available and used by WebSphere Partner Gateway. To avoid performance suffering, the pool size must provide sufficient number of connections for all WebSphere Partner Gateway components at all times. The following table describes the settings that can be used:

Data source Name	Description	Max	Min
datasources/bcgDocMgrDS	Used by BCGDOCMGR to connect to BCGAPPS database	120	1
datasources/bcgRCVRDS	Used by BCGRECEIVER to connect to BCGAPPS database	100	1
datasources/bcgConDS	Used by BCGCONSOLE to connect to BCGAPPS database	100	1
datasources/bcgMASDS	Used by BCGMAS to connect to MASDB database	100	1
Datasources/bcgArchiverDS	Used by the WPG Archiver application to connect to BCGAPPS database	100	1

Table 1 – Connection Pool Settings

Navigate to the following page from WebSphere Application Server admin console to set the connection pool properties mentioned in "Table 2 – Connection Pool Settings":

Resources > JDBC > Data sources > <JNDI name> > Connection pool properties

2.1.6 Statement cache

WebSphere Application Server has a single statement cache for each data source. In an application, the best performance can generally be achieved when the statement cache is large enough to hold all the prepared statements. If this cache is not large enough and is full, statements in the cache will be discarded to make room for newly prepared statements. If the discarded statements are needed at a later stage, then you will have to rework on the statements.

To set the statement cache size, navigate to the General Properties of the data source properties page.

Resources > JDBC > Data sources > (datasources/bcgDocMgrDS on bcgdocmgr and datasources/bcgMASDS on bcgmas for each host) > Additional Properties > WebSphere *Application Server data source properties > General Properties > Statement cache size.*

A cache size of 100 for all data sources is found to be optimal.

2.2 Java Virtual Machine settings

The Java Virtual Machine has multiple tuning parameters, which may be used to improve WebSphere Partner Gateway performance. The most important of these are related to garbage collection, setting the Java heap size, and configuring thread parameters.

You can find a detailed explanation of IBM JVM, along with a complete listing of all available parameters in the IBM JVM Diagnostic Guide. This guide can be found at http://www.ibm.com/developerworks/java/jdk/diagnosis/

2.2.1 Monitoring garbage collection

To set the heap size correctly, you must first determine the usage of the heap. This is easily done by collecting a verbosegc trace. A verbosegc trace prints garbage collection actions and statistics to standard error. To activate the verbosegc trace, use the Java run-time option - "-verbose:gc". For WebSphere Partner Gateway, this value is set via WebSphere Application Server admin console:

Servers > *Application servers* > <server> > *Java and Process Management* > *Process Definition* > *Java Virtual Machine*-> *Generic JVM Arguments.*

The verbosegc output would appear in "native_stderr.log" in the log subdirectory:

<WPG_install_dir>/wasND/Profiles/<profile name>/logs/<server name>/

The following section provides a sample of verbosegc output:

```
<af type="tenured" id="15" timestamp="Fri Jun 01 18:05:24 2007"
intervalms="15645.865">
  <minimum requested_bytes="24" />
  <time exclusiveaccessms="0.447" />
  <tenured freebytes="4294656" totalbytes="1073741824" percent="0" >
    <soa freebytes="0" totalbytes="1069447168" percent="0" />
    <loa freebytes="4294656" totalbytes="4294656" percent="100" />
  </tenured>
  <gc type="global" id="15" totalid="15" intervalms="15656.499">
    <refs_cleared soft="192" weak="601" phantom="0" />
    <finalization objectsqueued="5794" />
    <timesms mark="182.440" sweep="24.124" compact="0.000" total="206.892" />
    <tenured freebytes="996318152" totalbytes="1073741824" percent="92" >
      <soa freebytes="993097160" totalbytes="1070520832" percent="92" />
      <loa freebytes="3220992" totalbytes="3220992" percent="100" />
    </tenured>
  </gc>
  <tenured freebytes="996317632" totalbytes="1073741824" percent="92" >
    <soa freebytes="993096640" totalbytes="1070520832" percent="92" />
    <loa freebytes="3220992" totalbytes="3220992" percent="100" />
  </tenured>
  <time totalms="216.231" />
</af>
```

Each <af>…</af> section in the output actually indicates an invocation of the garbage collection (because of a memory allocation failure). The most interesting parts are the

“intervalms” attribute of element <af> and the “totalms” attribute of sub-element <time>. They represent the interval between the present and previous garbage collection invocation and the total execution time of this garbage collection respectively.
The ratio of `totalms/intervalms` is calculated to arrive at the time spent in garbage collection. Normally, the ratio is below 10%. Higher than the specified 10% implies that too much time is spent on garbage collection (and not on doing the actual work).

Another important metric in this example is to note the last <tenured> (after <gc>). It shows the status of the heap after the invocation of the specific garbage collection. This information helps us understand the heap usage of the application. This is useful for determining the appropriate heap size for the configuration.

Another helpful part is the sub element <timesms> of <gc>. It shows the time spent on the three phases of garbage collection: mark, sweep, and compact. Here, compact is the most important one and it should be zero most of the time. Try to avoid the compact phase as it the most expensive one. A high value usually means there are probably large memory allocations in the application. If possible, try to avoid large memory allocations.

2.2.2 Setting heap size

To determine the heap size, the first and the most important point is to ensure that the heap never pages. That is, the maximum heap size is contained in the physical memory.

Servers > *Application servers* > <server> > *Java and Process Management* > *Process Definition* > *Java Virtual Machine*:

For all WebSphere Partner Gateway components, 1536 MB is an optimal size for both Initial Heap Size and Maximum Heap Size.

2.2.3 Setting AIX threading parameters

The IBM JVM threading and synchronization components are based on the AIX POSIX compliant pthread implementation. The following environment variables have been found to improve Java performance in many situations, and are recommended for WebSphere Partner Gateway on AIX platform.

```
export AIXTHREAD_COND_DEBUG=OFF
export AIXTHREAD_MUTEX_DEBUG=OFF
export AIXTHREAD_RWLOCK_DEBUG=OFF
export AIXTHREAD_SCOPE=S
export SPINLOOPTIME=2000
```

These variables control the mapping of Java threads to AIX threads, turn-off mapping information, and allow spinning on mutex locks.

For more information on AIX-specific Java tuning, see:

- http://www.ibm.com/developerworks/eserver/articles/JavaPart1.html
- http://www.ibm.com/developerworks/eserver/articles/JavaPart2.html

2.3 Network settings

The available parameters and their default settings vary with operating system and also with the choice of physical medium.

2.3.1 AIX

Interface-Specific Network Options (ISNO) allows IP network interfaces to be custom-tuned for best performance. The following is a good value for ISNO options of TCP/IP interface for all systems:

```
chdev -l en0 -a tcp_recvspace=65536 -a tcp_sendspace=131072 -a tcp_nodelay=1
```

tcp_recvspace" parameter controls the number of bytes of data that the receiving system can buffer in the kernel on the receiving sockets queue. It is also used by the TCP protocol to set the TCP window size. The TCP uses the TCP window size to limit the number of bytes of data while sending to the receiver. This will ensure that the receiver has enough space to buffer the data. A common guideline for the tcp_recvspace tunable is to set it to a value that is at least 10 times less than the MTU(Maximum Transfer Unit) size.

"**tcp_sendspace**" parameter controls the quantity of data that the sending application can buffer in the kernel before the application is blocked on a send call. Set it at least as large as the value of "tcp_recvspace". For higher speed adapters, the "tcp_sendspace" value is at least twice the size of the "tcp_recvspace" value.

"**tcp_nodelay**" parameter controls whether to remove the delay caused by Nagle Algorithm mixed with TCP delayed ACK (by enabling this option).

2.4 WebSphere Application Server settings

WebSphere Partner Gateway uses WebSphere Platform Messaging for JMS communication.

2.4.1 Messaging engine data store database tuning

With WebSphere Partner Gateway, WebSphere Platform Messaging uses a database for storing the messages it processes. The messaging database, MASDB is created for this purpose. The tuning consideration discussed in Application database tuning is also applicable here.

It is recommended to disable the DB2 auto-runstats facility for MASDB. Sometimes, a "spike" load on WebSphere Partner Gateway can make the system unstable.

In case of the Simple Mode setup, it is recommended to place the file store, used for storage, on a fast disk system.

2.4.2 Setting messaging engine buffer size

Every Messaging Engine manages two memory buffers for messages and message-related data. The size of these memory buffers is modified to achieve better performance.

The memory buffer size can be set by navigating to the Custom Properties page of a messaging engine.
On the WebSphere Application Server Admin Console, go to *Service integration > Buses > BCGBus > Messaging engines > bcgmasCluster.000-BCGBus > Custom properties.*

Add the following properties along with their given values (by default these are not present):

- sib.msgstore.cachedDataBufferSize:60000000
- sib.msgstore.discardableDataBufferSize:20000000

2.4.3 WebSphere Thread Pool Configuration

The WebSphere Thread Pools are used to allocate threads for processing the different components.
It is recommended to set the Max Thread Pool Size to 60 for the Default Thread Pool. Navigate to *Application servers* > <server> > *Thread Pools* > *Default* and set the Maximum Size to 60.

2.5 File system tuning – NFS settings

NFS is used when the components need to share the Common File System (as in the distributed mode or when multiple instances of the product are installed). In such cases, the Common File System is first locally mounted on a machine (For NFS server, use the machine hosting the BCGRECEIVER). The BCGDOCMGR then remotely mounts the Common File System via NFS. The following sections describe how to use the tuning parameters related to NFS with WebSphere Partner Gateway.

2.5.1 AIX

For more information on NFS tuning with AIX, see pSeries and AIX Information Center: NFS performance.

NFS server side:

1. The following nfso parameters can be used:
   ```
   nfs_rfc1323=1
   nfs_tcp_socketsize=600000
   nfs_max_threads=3891
   nfs_max_connections=0
   ```
2. In /etc/exports, add the following parameters:

   ```
   /common *(rw,sync,no_acl,no_subtree_check,no_root_squash)
   ```

NFS client side:

1. The following nfso parameters can be used:
   ```
   nfs_rfc1323=1
   nfs_tcp_socketsize=600000
   ```
2. In /etc/filesystems, add the following parameter to mount remote CFS via NFS:

   ```
   /common:
   dev = "/common"
   vfs = nfs
   nodename = <node address>
   mount = false
   options =
   bg,hard,intr,noacl,biods=128,rsize=65536,wsize=65536,actimeo=300,sec=sys
   account = false
   ```

2.5.2 Linux

NFS server side:

1. Use 32 threads for kernel nfs server (modify the parameter 'USE_KERNEL_NFSD_NUMBER' in /etc/sysconfig/nfs).
2. In /etc/exports, add the following parameters:

   ```
   /common *(rw,sync,no_acl,no_subtree_check,no_root_squash)
   ```

NFS client side:

1. In /etc/fstab, add the following parameter to mount remote CFS via NFS:

   ```
   <node >:/common /common nfs
   nfsvers=3,noatime,rsize=65536,wsize=65536,intr 0 0
   ```

2.6 WebSphere Partner Gateway specific tuning considerations

The subsequent sections describe the various tuning considerations of WebSphere Partner Gateway.

2.6.1 Common File System considerations

WebSphere Partner Gateway components use a common shared storage to facilitate the processing and storage of the documents. The usage of the common file system is quite extensive in document processing, that is, for storing intermediate documents, for non-repudiation store, for message store, and so on.

If either the non-repudiation or the message store functionality is not required, it can be turned off for utilization of disk space.

For improved performance, place the common file system storage on a fast disk subsystem. This will reduce the disk I/O time and thereby the wait time of the processor for I/O operations to complete.

2.6.2 BPE thread tuning considerations

The Business Process Engine (BPE) is at the heart of the WebSphere Partner Gateway system and is responsible for most of the processing. It is imperative that the BPE Threads are tuned appropriately to achieve the best throughput performance.

Three types of BPE beans: namely, Main, Signal, and Sync are deployed in the system for processing documents as Message Driven Beans (MDB). For these MDBs, the BPE Threads are set by modifying the “Maximum Concurrent Endpoint” setting of specific JMS Activation Specification.
To change the BPE threads, do the following:

1. In the WebSphere Application Server Admin Console, navigate to *Resources > JMS > Activation Specifications*
2. *S*elect the specific Activation Specification and change the “Maximum Concurrent Endpoint” value.

Maintain a balance in setting the Maximum Concurrent Endpoints. Ensure it is high enough to allow a higher throughput, and at the same time does not cause wait time issues.

Although it is very specific to the environment and platforms on which WebSphere Partner Gateway is executing, the number of “Maximum Concurrent Endpoints” can be nearly three to four times than that of the number of cores on which the Document Manager server is running.

2.6.3 Delivery Manager tuning considerations

The Delivery Manager (DM) is a component of the Document Manager that is responsible for delivering the documents to trading partners.

The DM comprises of the Gateway Threads, which can be set for a particular destination from the WebSphere Partner Gateway Console.

For a better throughput performance, the destination gateway threads should deliver continuously. Specifically, in case of a Multiple Document Manager deployment, the Delivery Manager for all Document Managers instances should be delivering documents for a better throughput.

In the WebSphere Partner Gateway console, navigate to *System Administration > DocMgr Administration > Delivery Manager*.

- The *bcg.delivery.allowDMLoadBalance* parameter is used in Horizontal Scaling scenarios. This parameter controls the setting that allows multiple Delivery Managers to deliver documents to the partners concurrently. If this parameter is set to "No", then at any time, only one DM is allowed to work on a specific destination. It is recommended to set this value to "Yes".

It is important to monitor the Destination Queue so that it does not become a bottleneck for the whole system. The Destination Queue is a File System queue located on the Common File Systems and is unique to each Destination created for a partner. For best performance, the destination queue has to be bounded and cannot grow alarmingly. In WebSphere Partner Gateway console, navigate to *Viewers > Destination Queue* to monitor the Destination Queue.

If this queue accumulates, add more Gateway Threads and see if it helps.
To update Gateway Threads from the WebSphere Partner Gateway Console:

1. Navigate to the *Destination Details (Profiles > <Partner> > Destination Details > <Destination>)* page of a Partner
2. Change the "Number of Threads" parameter.

Note: If the value of the Gateway Threads is too high, then it can slow down the performance. The number of Gateway Threads can be nearly two times than that of the number of cores on which the Document Manager server is running.

2.6.4 Event Engine tuning considerations

The Event Engine is a component of WebSphere Partner Gateway. It is responsible for logging events associated with the document processing to WebSphere Partner Gateway database.

Although the event logging is an asynchronous operation, there is always an overhead on the database to log multiple events associated with the processing.

The events associated with document processing are sent to the DATALOGQ jms queue for asynchronous processing. For a typical AS2 document processing test, there are four events that are logged. The document itself is logged to this queue multiple times as the state of the document changes.

With the given amount of processing, configure the Event Engine appropriately to balance the document processing time and the time it takes for the document events to show up on the console. The DATALOGQ has to be bounded and cannot grow alarmingly.
It is recommended that the number of Event Engine threads (the Maximum Concurrent Endpoint setting for DATALOGQ) be equal to the number of BPE threads.
To change the setting of Maximum Concurrent Endpoint from the Admin console of WebSphere Application Server, navigate to
Resources > JMS > Activation Specifications > jms.bcg.as.datalogQAS.

Exclude some debug and informational events

During document processing, WebSphere Partner Gateway will log various events, which are then displayed on the WebSphere Partner Gateway Console. These events are: debug, informational, or indicate an error in the document processing. If you want to disable logging of certain debug and informational events, you can configure the same in the console.

A WebSphere Partner Gateway administrator has the choice to log fewer events, thereby reducing database processing time. The administrator can set the *bcg.event_log_exclude* parameter from the console. To access this attribute, navigate to *System Administration > Common Properties*.

The administrator can then set the *bcg.event_log_exclude* to exclude certain known events by providing comma separated event codes.

2.6.5 State Engine and other threads considerations

WebSphere Partner Gateway has state engines for AS, RosettaNet, and ebMS document types, which run as background threads. These state engines run as scheduled threads and are responsible for managing states as defined in the specifications of the respective protocols.

The State Management is essentially done by polling the database at specific intervals for any new activity.

If WebSphere Partner Gateway is not processing a document flow for a specific type, the state engine thread for that document type can be switched off to reduce the database access.

The following thread counts can be set to zero for the specific document type as shown below:

Document Type	Parameter
AS	System Administration DocMgr Administration > AS State Engine bcg.asstate.thread_count
ebMS	System Administration DocMgr Administration > Process Admin > Reliable Messaging bcg.rm.pollInterval
RosettaNet	System Administration > DocMgr Administration > RosettaNet bcg.rne.inbound_poll_interval

Table 3 – State Engine settings

It is recommended to set the interval of the AS State Engine thread to a value of 30000. The default value is 1000. To navigate to this attribute, go to *System Administration > DocMgr Administration > AS State Engine > bcg.asstate.runinterval.*

Apart from the State Engines, there are certain other background threads that poll the database for specific activities. These threads are for the Summary Engine, Sponsor Engine, and Archiver. Unless specifically required, it is not recommended to change the default values for these threads.
To change the default value of these threads, navigate to *System Administration > DocMgr Administration > Others* in the Console of WebSphere Partner Gateway.

2.6.6 Non-repudiation considerations

By default, WebSphere Partner Gateway stores inbound and outbound documents in the non-repudiation store of the Common File System. The purpose of non-repudiation as summarized in the WebSphere Partner Gateway documentation is “Digital signing is the mechanism for ensuring non-repudiation. Non-repudiation means that a participant cannot deny having originated and sent a message. It also ensures that the participant cannot deny having received a message.”

Non-repudiation will add some overhead to document processing, as both inbound and outbound documents will be stored as-is in the non- repudiation store of the Common File System, thereby increasing the I/O time.

Turning off non-repudiation

In WebSphere Partner Gateway, you can turn off or turn on non-repudiation for a specific channel between trading partners. This can be accomplished by setting the *"Non-Repudiation Required"* parameter to "No".

The "*Non-Repudiation Required*" is available as a channel attribute when you create a channel between the source and destination partners from the console.

2.6.7 Data Archival and Purge

As WebSphere Partner Gateway processes documents over a period of time, data grows in the WebSphere Partner Gateway application database and the Common File System. It is required and also advised to purge data regularly for maintenance and performance. WebSphere Partner Gateway Archiver is a tool that provides the mechanism for archiving and purging such old data.

The Archiver operation is database intensive and can affect the overall system throughput. It is recommended that the Archiver be scheduled to run when WPG is not processing documents or the processing volume is low.

The following tuning parameters are available for the WebSphere Partner Gateway archiver and can be set from the *System Adminstration > DocMgr Administration > Others > Archiver* page of WebSphere Partner Gateway Console:

- *bcg.archive.db.rowsPerBatch* – batch of data used by temporary tables during purging. Larger batch size helps to reduce the overheads of temporary tables. The default value is 100000.
- *bcg.archive.db.rowsPerStatement* – delete chunk size used for purging from the batch. Larger delete sizes can cause slow purging. The default value is 1000.
- *bcg.archive.db.parallelism* – parallelism for deleting data from tables. Larger parallelism helps improve the utilization of DB server. However, too many threads can thrash the DB Server. If the number of DB cores is X, then the recommended value is around X/2. The default value is 4.
- *bcg.archive.activityParallelism* – parallelism used for the copy/purging activity. The default value is 4, which is also the maximum allowed value.
- *bcg.archive.maxThreads* – number of threads allocated to copy/delete for the common file system data. The default value is 4.

Note that there are no upper or lower bounds for the values unless explicitly mentioned. The default values are recommended for most deployments.

3 Appendix

3.1 Appendix A – Terminologies

Some terminologies used in this guide.

- Common File System

WebSphere Partner Gateway uses a file system to store process documents, non repudiation documents, message store documents, and any intermittent temporary files created during the processing of the documents. This file system is shared by all the WebSphere Partner Gateway components and is called the Common File System.

- BCGAPPS

BCGAPPS is the WebSphere Partner Gateway Application Database.

- BCGCONSOLE

This is the application server that runs the WebSphere Partner Gateway console application bcgConsole.

- BCGDOCMGR

This is the application server that runs the WebSphere Partner Gateway Document Manager applications; the bcgBPE and bcgDocMgr applications.

- BCGMAS

BCGMAS is the WebSphere Partner Gateway Messaging Application Server.

- BCGRECEIVER

This is the application server that runs the WebSphere Partner Gateway Receiver application bcgReceiver.

- MASDB

The Simple Distributed mode and Full Distributed mode deployment scenarios use the BCGMAS. The BCGMAS server makes use of a database for persistent storage. This database is called as the MASDB.

- Throughput

  Throughput is defined as the number of business documents processed per second.