



Performance Tuning Guide

Note

Before using this information and the product it supports, read the information in "Notices" on page 31.

This edition applies to version 3.6.0 of IBM FileNet Business Process Framework (product number 5724-R75) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2006, 2009.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Table of contents

Table of contents	5
Introduction	7
Objective	7
BPF configuration tuning	8
Enabling open Case event	8
Disabling attachment properties update	8
Disabling attachment list	8
Disabling Inbasket case count	8
Filtering out locked workobjects	8
Role management configuration	9
Application settings – Metadata cache	9
What information is cached in BPF?	9
Enabling the metadata cache	9
Estimating the size of the BPF metadata cache.....	10
Administering the BPF metadata cache	11
BPF metadata cache and locales.....	11
Application settings – Enable profiling	11
Application settings - XMLHTTP logging	11
Application settings - Case ID Manager reservation size	12
Application settings – Folder to file case objects	12
Application settings – Inbasket browse list page size.....	12
Application Settings – Case attachments list – Adhoc search definition	12
Case fields – Lookup fields	16
Web Application logging.....	16
Avoid large dynamic pick lists	18
Avoid large numbers of Read From Workflow fields.....	18
Annotation viewing	18
Bulk Processing	18
Disable Inbasket(s) List on Search	18
Content Engine tuning	19
Query Options	19
Disable trace logging.....	20
Index Content Engine Objects	20
Process Engine tuning	22
Purge workflow event log	22
Disable unnecessary events	22
Create indices	22
Turn off VWRouter logging	22
Monitor number of server processes	22
Web Application container tuning.....	23
Internet Explorer settings	23
Cache settings.....	23
BEA WebLogic	23
Web container thread pool settings.....	23
Connection pool size	24
Heap Size	24
JIT compiler settings.....	25
HTTP connection timeout.....	26

Table of contents

IBM WebSphere	27
Servlet caching.....	27
Web container thread pool settings	27
Connection Pool size	28
Heap size	28
JIT compiler settings	29
Notices	31
Trademarks.....	32

Introduction

BPF is a Case Management framework application. In contrast to BPM, which typically operates with only a single workflow object, or ECM, which operates with a single document or piece of content when manipulated through Workplace, BPF operates with complex business objects. A BPF business object (referred to as a Case) is a collection of custom objects in a CE repository, which include the Case object, attachment objects (documents, folders, or custom objects), audit trail event objects, etc. A BPF Case may or may not be related to one or more workflow objects in the PE as well. BPF always operates Case in terms of BPF Cases, although it could be configured to mimic BPM step processor behavior. BPF presents its business objects (Cases) in a consistent and very intuitive manner, hiding the complexity of the different components stored and managed in different repositories, allowing for synchronization of the individual Case, its attachments (both documents and custom objects) and workflow. BPF transactions are significantly more complex than the typical BPM or ECM transactions, as one BPF transaction initiates multiple object transactions in multiple products and repositories.

The key components of BPF are its configurable user interface (the BPF Web Application) and a framework for background step processing.

The BPF Web Application is built around the AJAX (Asynchronous JavaScript™ and XML) development model and makes extensive use of the IBM® FileNet® P8 Platform Web Application Toolkit (WAT) as well. The BPF Web Application exposes Case Management services via an action dispatcher servlet. The Web client uses XMLHTTP (AJAX) for communication with the action dispatcher. The design of the BPF Web Application minimizes the client-server traffic by updating the browser only with the data that has changed. The BPF Web Application also contains a set of client-side JavaScript functions that employ data validation techniques, minimizing unnecessary round trips to the web server.

The BPF Web Application user interface layout is metadata driven and is built dynamically upon user request. The metadata describing the user interface layout is stored in the BPF Metastore database and is configured via the BPF Explorer administrative tool. When the user makes a request to retrieve business objects, the business object presentation manifest and any associated business rules are retrieved from the BPF Metastore, packaged as XML, and then sent to the browser, where the proper HTML is rendered via client-side XSL transformation. The BPF Metastore Cache Manager virtually eliminates the database traffic by caching all frequently used presentation manifest information.

The BPF background step processor framework is implemented as a FileNet P8 Platform Component Integrator Java™ adaptor. This allows BPF operations to be placed as steps on the process map. Each operation is configurable and allows the workflow designer to implement both simple and relatively complex background processes with no developer assistance required. The key operations available in BPF include: Create Case, Attach Document, Update Case, and Log Event. Note that Create Case and Attach Document both support the attachment of custom objects as well as documents.

Objective

This section explains some of the key performance-related aspects of BPF. It includes a high-level description of the BPF architecture so as to provide a better understanding of the performance and scalability related aspects of BPF as well as of the tuning parameters that can impact its performance.

NOTE It is recommended that system administrators implementing BPF first review and implement recommendations published in the *IBM FileNet P8 Platform Performance Tuning Guide* to tune the baseline FileNet P8 Platform environment before tuning BPF.

BPF configuration tuning

BPF has many useful features that add tremendous value to the application but deploying these may also add considerable performance overhead. It is important to weigh the value provided by these features against the performance impact that will be incurred. These tunable options are described below with some general indications of the performance impact that they can have.

Enabling open Case event

The Enable Open Case Event option in the Inbasket Configuration – General Tab, provides additional audit logging each time the Case is opened. If this option is unchecked, BPF only logs the action entry when the Case is dispatched, not the entry time when the Case was opened. The Enable Open Case event option is somewhat expensive, involving the creation of additional AuditLog item entries. Leaving it unchecked will result in a performance gain of perhaps several hundred milliseconds.

Disabling attachment properties update

When a Case has attached content, it is possible to update one or more attached content properties when the Case properties have changed and the Case object is updated. This adds overhead to the action of saving a Case. The Disable Updating Attachment Properties option (in the Options pane on the right-hand side of the General tab of the Inbasket Configuration dialog for each Inbasket) can, if checked, improve performance considerably since updating attachment properties is an expensive transaction, and the expense, in terms of slower performance, increases in direct proportion to the number of attachments each Case has.

Disabling attachment list

Generally, attached content is shown in the lower part of the Case tab pane. This adds overhead in loading the Case tab because BPF needs to establish the hyperlinks to the attachments as well. Since the Case tab is typically the default display tab, the wait time may be more noticeable to the user. Attachments can also be displayed via the Attachments tab. If possible, set the Attachment Visibility option dropdown value (in the Options pane on the right-hand side of the General tab of the Inbasket Configuration dialog for each Inbasket) to No and consider using the separate Attachments tab instead. This will allow the Case tab to load much more quickly since it will not have to load hyperlinks to the attachments as well.

Disabling Inbasket case count

This feature provides an Inbasket count, which is displayed immediately following the Inbasket name in the Web Application but is controlled at the role level. Consider deselecting the Display Case Count checkbox on each security profile and exposing the queue_depth tool (for use as needed) instead. Otherwise, BPF will have to recalculate the Case count each time a Case is dispatched or an Inbasket filter value changed and the open Inbasket clicked on, which is a very expensive operation.

Filtering out locked workobjects

The filters determine how the workflow queue will be browsed in order to create the Inbasket view. Each filter has modifiable SQL syntax behind it. These are defined in the Workflow Configuration section of BPF Explorer and then selected in the Filter Name dropdown in the Inbasket Configuration – General tab. The default filters pre-defined in BPF Explorer show both locked and unlocked work objects. By contrast, it can be far less expensive to browse a queue and filter out any items locked by other users. To add this additional SQL parameter, modify the appropriate filter in BPF Explorer by adding a term such as (F_LockUser = %USERID% or F_LockUser = 0) to the WHERE clause as a way of filtering out any items locked by users other than the current user.

For example:

```
select %PAGESIZE% F_WobNum, F_UniqueId, F_Locked, F_LockUser, F_BoundUser,  
Bp8CaseID, 9999 as queue_depth from %VIEWNAME% WHERE %FILTERBY% (F_LockUser =  
%USERID% or F_LockUser = 0) and F_StepName='%STEPNAME%' order by %ORDERBY%,  
F_EnqueueTime, F_UniqueId asc
```

It is much less expensive to handle this kind of filtering at the SQL level than at the ResultSet level, after the fact.

Role management configuration

BPF supports three different models of role management configuration. They are:

- **Manual** – Role membership is assigned and managed via BPF Explorer only.
- **LDAP group integration** – Role membership is managed via LDAP security group membership. BPF roles are linked to LDAP groups.
- **Workplace access roles** – Role membership is managed via LDAP security group membership. BPF roles are linked to Workplace access roles, which are linked to LDAP groups.

For reasons of performance enhancement, it is strongly recommend that you use Workplace access roles instead of direct LDAP integration, especially in cases where the LDAP directory is large or complex (containing many levels of nested groups/folders).

Application settings – Metadata cache

You can enable caching of the BPF metadata to reduce the number of database transactions that BPF makes when the BPF Web application requires configuration information. When you enable metadata caching, BPF caches the metadata the first time a user performs an action that retrieves the configuration settings from the BPF Metastore.

What information is cached in BPF?

BPF stores the configuration settings, or metadata, for inbaskets, cases, and related items in the BPF Metastore database. This metadata includes, but is not limited to:

- The layout that is used when a user opens a case from an inbasket
- The names and other properties of the case fields that are to be displayed when a user opens a case
- The names and other properties of the case fields that are to be displayed when a user clicks an inbasket to display a case in browse mode
- The list of inbaskets along with the relevant configuration information for these inbaskets that a user can access based on that user's BPF role

Enabling the metadata cache

You use BPF Explorer to enable caching of BPF metadata.

To enable caching of BPF metadata:

1. Log in to BPF Explorer.
2. In the console tree, click **Application Settings > System-wide Settings**.
3. In the details pane, set the **Enable BPF metadata cache** to True.

Estimating the size of the BPF metadata cache

You can estimate the size of the metadata cache required for your application. This information is particularly useful when you are designing a large application that employs more than ten case types, hundreds of inbaskets, and tens of BPF roles. Estimating the cache required can help you prepare the hardware infrastructure and tune the Web application container needed to support your application.

TIP The calculations in the following topics assume that you are using BPF 3.6.0-004 or higher.

Determining the number of items in the BPF Metastore

You can use SQL queries to determine the number of items in the BPF Metastore. This value provides an understanding of the quantity of items that might need to be cached.

To determine the number of items in the BPF Metastore:

Determine the number of items in the BPF Metastore by using the following queries:

To determine the number of . . .	Use this query . . .	Case Management sample application
Case types (Nc)	<code>select count(*) as Nc from CASE_TYPES</code>	Nc = 1
BPF roles (Np)	<code>select count(*) as Np from SECURITY_PROFILE</code>	Np = 5
Inbaskets (Ni)	<code>select count(*) as Ni from INBASKET_CONFIG</code>	Ni = 14
Role and inbasket combinations (Npi)	<code>select count(*) as Npi from INBASKETS_ROLES</code>	Npi = 14
Tools (Nt)	<code>select count(*) as Nt from TOOLS</code>	Nt = 9
Inbasket functions (Nf)	<code>select count(*) as Nf from INBASKET_FUNCTIONS</code>	Nf = 41

4. Calculate the total number of items in the BPF Metastore by using the following formula:

$$N = (6 + Nt + Nf) * Nc + (3 * Np) + (5 + 2 * Nc) * Ni + (2 * Npi) + (4 * Nf)$$

For example, the number of items in the BPF Metastore for the sample Case Management application is:

$$N = (6 + 9 + 41) * 1 + (3 * 5) + (5 + 2 * 1) * 14 + (2 * 14) + (4 * 41) = 361$$

Estimating the memory size of the fully-loaded metadata cache

To estimate the memory size required for your metadata cache when it is fully loaded:

Export the manifest XML file for your application and determine the size of the file (Fs) in megabytes.

Use the following formula to determine the memory size in megabytes of the fully-loaded BPF cache:

$$\text{Memory} = Fs * 20$$

For example, the manifest XML file for the sample Case Management application is 0.35 MB. Therefore, the estimated memory size for the fully-loaded metadata cache is:

$$\text{Memory} = .35 * 20 = 7 \text{ MB}$$

Administering the BPF metadata cache

You can administer the BPF metadata cache by using the BPF cache management Web page. From this Web page, you can

- Review the contents of the metadata cache
- Enable or disable the metadata cache
- Refresh or clear the metadata cache

To access the BPF cache management Web page, use the following URL:

```
http:// host-name:port/bpf-app/Bp8Admin.jsp
```

Where:

- *host-name* is the name of the server on which your BPF application resides.
- *port* is the port number for the server.
- *bpf-app* is the name of your BPF application.

You must log in using a user account that belongs to the Application Engine Administrators group.

BPF metadata cache and locales

The BPF metadata is specific to the user's locale as determined by the language setting in that user's Web browser. If a configuration item is required by users in two different locales, the cache contains two copies of the item — one for each locale.

For example, assume that your application is used by two users, one who's locale is set to English (United States) and one who's locale is set to French (France). Both users access an inbasket with the identifier *xx*. As a result, BPF creates two items in the metadata cache for the inbasket. These items have the following keys:

```
en_US INBASKETS_FUNCTIONS (inbasketId:xx)  
fr INBASKETS_FUNCTIONS (inbasketId:xx)
```

When you are planning for your BPF metadata cache, you must consider the impact of locales on the cache size.

Application settings – Enable profiling

This should be set to False. Setting the Troubleshooting – Enable Profiling application setting (under the Web Application group) to True will allow `log4j.xml` settings higher than Error (such as Debug for example) to result in the additional information being written out to the BPF log files so configured in the `<web_root>\WEB-INF\log4j.xml` file. This allows logging (other than for errors) to be turned off very easily, even when the logging level in `log4j.xml` is set to Debug, without modifying `log4j.xml`. Nevertheless, for performance reasons, it is recommended that this application setting be set to False unless debug-level logging is considered absolutely necessary, as debug-level logging is extremely expensive in terms of performance.

Application settings - XMLHTTP logging

This should be set to False. Another logging-related BPF application setting that can affect performance is Enable XML action messages logging (under the XML Message Logging group). Setting this to True can be very useful for debugging purposes, but is extremely harmful in terms of performance. It is strongly recommend that this setting be set to False in production environments.

Application settings - Case ID Manager reservation size

This should be set relatively high (greater than or equal to the default of 100). The BPF Case ID Manager reservation size application setting (under the System-wide Settings group) is used by both the BPF Web Application and by BPF_Operations (although independently of one another) to determine the number of Bp8CaseID values the application should reserve and cache each time it runs out of previously reserved and cached values and needs to create a new BPF Case. The lower this value is set, the more frequently BPF will need to execute the code that reserves and caches a range of Bp8CaseID values. Since this is an expensive operation, we strongly recommend setting this value higher, rather than lower, when configuring a production system.

Application settings – Folder to file case objects

It is recommend you leave the Object Store Folder to File Case Objects application setting (under the System-wide settings group) blank. BPF allows specifying a CE repository folder where the Bp8Case, Bp8AuditTrail, and Bp8Attachment objects instances are filed. While it is sometimes useful to have all these objects filed in a particular folder, there is a performance penalty, which may result in significant performance degradation as number of objects increases. It is recommended that you disable filing Case objects in a folder by keeping the setting blank.

Application settings – Inbasket browse list page size

It is recommended that you keep the Inbasket Page Size application setting (under the Web Application group) set to a value less than 100. BPF allows specifying a number Case objects that will be displayed on a single page in the Inbasket browse list. It is recommended you keep page size under 100 records to avoid unnecessary load on the server.

Application Settings – Case attachments list – Adhoc search definition

The CE SQL query that is installed by default for this setting, which builds the list of attached documents and custom objects that appears as icons at the bottom of the Case tab, is configured to be as broad as possible, taking into account the possibility that the customer might be storing attached documents and custom objects in CE folders and being, of necessity, unaware of the specific document and custom object subclass names being used for attachments at any given customer site. As a result, the default query references the entire document and customobject pseudo-tables and joins these results with the folder pseudo-table, which can result in performance issues if a customer has many other document and custom object subclasses and instances in their system than those actually being used as attachments with BPF.

One way of addressing this issue is for customers to customize CE SQL query (configured in BPF Explorer under **Applications Settings > Web Application**) in a customer-specific fashion to provide improved performance.

NOTES

- BPF does not support any such customizations beyond providing examples, such as the following one, of how this might be done in principle.
- Other, similar, queries for the Attachment tab and Viewer Documents List can also be found in BPF Explorer under **Applications Settings > Web Application** as well, and could potentially be customized in similar ways.

The following is an example of how this query might be customized to improve performance.

By default, the DOCUMENT_LIST_PANEL_SQL CE query looks like this:

```
select a.Bp8ObjectID as Id, a.Bp8ObjectType as ObjectType, a.Id as AttachID,
d.DocumentTitle, d.VersionSeries, d.VersionStatus, d.MajorVersionNumber,
d.MinorVersionNumber, d.MimeType, d.isReserved, d.IsVersioningEnabled,
f.FolderName, c.Bp8CaseID from ((Bp8Attachment a left join Document d ON
d.VersionSeries = a.Bp8VersionSeries) left join Folder f ON f.Id =
a.Bp8ObjectGUID) left join CustomObject c ON c.Id = a.Bp8ObjectGUID where
a.Bp8CaseID = ^1 and ^2 order by a.Bp8ObjectType desc, f.FolderName asc,
d.DocumentTitle asc
```

In the case of an actual BPF Case with Bp8CaseID 66746 and two attachments (one a Document and one another Case object), the ^1 and ^2 are being replaced to produce the following actual CE query:

```
select a.Bp8ObjectID as Id, a.Bp8ObjectType as ObjectType, a.Id as AttachID,
d.DocumentTitle, d.VersionSeries, d.VersionStatus, d.MajorVersionNumber,
d.MinorVersionNumber, d.MimeType, d.isReserved, d.IsVersioningEnabled,
f.FolderName, c.Bp8CaseID from ((Bp8Attachment a left join Document d ON
d.VersionSeries = a.Bp8VersionSeries) left join Folder f ON f.Id =
a.Bp8ObjectGUID) left join CustomObject c ON c.Id = a.Bp8ObjectGUID where
a.Bp8CaseID = 66746 and (((VersionStatus = 1 OR VersionStatus is null) AND
d.isCurrentVersion = True) OR a.Bp8ObjectType = 15) order by a.Bp8ObjectType
desc, f.FolderName asc, d.DocumentTitle asc
```

In the CE Catalog trace log, this CE SQL query is being converted to actual Microsoft® T-SQL as follows by CE itself before being executed against the object store database:

```
2006/02/25 00:55:34.546 0750 0034 Text0<SELECT T1.u41_bp8objectid,
T1.u42_bp8objecttype, T1.object_id, T2.u2e_documenttitle, T2.version_series_id,
T2.version_status, T2.major_version_number, T2.minor_version_number,
T2.mime_type, T2.is_reserved, T2.versioning_enabled, T3.name, T4.u3d_bp8caseid,
T1.object_class_id, T1.security_id, T2.object_class_id, T2.security_id,
T3.object_class_id, T3.security_id, T4.object_class_id, T4.security_id

FROM    Generic T1

        left outer join (SELECT * FROM DocVersion D2 WHERE D2.home_id IS NULL) T2
on ( T2.version_series_id = T1.u47_bp8versionseries )

        left outer join (SELECT * FROM Container D3 WHERE D3.home_id IS NULL) T3
on ( T3.object_id = T1.u45_bp8objectguid )

        left outer join (SELECT * FROM Generic D4 WHERE D4.home_id IS NULL) T4 on
( T4.object_id = T1.u45_bp8objectguid )

WHERE T1.home_id IS NULL

AND T1.object_class_id IN (?)

and ( ( T1.u3d_bp8caseid = ? ) and ( ( ( ( T2.version_status = ? ) or (
T2.version_status is null) ) and ( T2.is_current = 1 ) ) or (
T1.u42_bp8objecttype = ? ) ) )

ORDER BY T1.u42_bp8objecttype DESC, T3.name , T2.u2e_documenttitle >

2006/02/25 00:55:34.546 0750 0034 InBindings0<{1B2F8C97-5D9F-4099-98C9-
6F4099A03359}, 66746, 1, 15>
```

The following is an example of a customized version of this query, which:

- Assumes that the customer in question is using CustomObjects as attachments

```
select a.Bp8ObjectID as Id, a.Bp8ObjectType as ObjectType, a.Id as AttachID,
d.DocumentTitle, d.VersionSeries, d.VersionStatus, d.MajorVersionNumber,
d.MinorVersionNumber, d.MimeType, d.isReserved, d.IsVersioningEnabled,
f.FolderName from (Bp8Attachment a left join Document d ON d.VersionSeries =
a.Bp8VersionSeries) left join Folder f ON f.Id = a.Bp8ObjectGUID where
a.Bp8CaseID = ^1 and ^2 order by a.Bp8ObjectType desc, f.FolderName asc,
d.DocumentTitle asc
```

In the case of an actual BPF Case with Bp8CaseID 66746 and two attachments (one a Document and one another Case object), the ^1 and ^2 are being replaced to produce the following actual CE query:

```
select a.Bp8ObjectID as Id, a.Bp8ObjectType as ObjectType, a.Id as AttachID,
d.DocumentTitle, d.VersionSeries, d.VersionStatus, d.MajorVersionNumber,
d.MinorVersionNumber, d.MimeType, d.isReserved, d.IsVersioningEnabled,
f.FolderName from (Bp8Attachment a left join Document d ON d.VersionSeries =
a.Bp8VersionSeries) left join Folder f ON f.Id = a.Bp8ObjectGUID where
a.Bp8CaseID = 66746 and (((VersionStatus = 1 OR VersionStatus is null) AND
d.isCurrentVersion = True) OR a.Bp8ObjectType = 15) order by a.Bp8ObjectType
desc, f.FolderName asc, d.DocumentTitle asc
```

This query still returns any custom objects that might be attached to Case 66746, but without the extra join to the CustomObject virtual table (and thus without the additional information such as c.Bp8CaseID returned by the default query). As long as custom objects are not being used as attachments, this should be no problem, and the resulting query should be faster because it avoids the extra join.

In the CE Catalog trace log, this CE SQL query is being converted to actual Microsoft T-SQL as follows by CE itself before being executed against the object store database:

```
2006/02/25 01:04:00.250 0984 0230 Text0<SELECT T1.u41_bp8objectid,
T1.u42_bp8objecttype, T1.object_id, T2.u2e_documenttitle, T2.version_series_id,
T2.version_status, T2.major_version_number, T2.minor_version_number,
T2.mime_type, T2.is_reserved, T2.versioning_enabled, T3.name,
T1.object_class_id, T1.security_id, T2.object_class_id, T2.security_id,
T3.object_class_id, T3.security_id
FROM   Generic T1
       left outer join (SELECT * FROM DocVersion D2 WHERE D2.home_id IS NULL) T2
on ( T2.version_series_id = T1.u47_bp8versionseries )
       left outer join (SELECT * FROM Container D3 WHERE D3.home_id IS NULL) T3
on ( T3.object_id = T1.u45_bp8objectguid )
WHERE  T1.home_id IS NULL
       AND T1.object_class_id IN (?)
and ( ( T1.u3d_bp8caseid = ? ) and ( ( ( T2.version_status = ? ) or (
T2.version_status is null) ) and ( T2.is_current = 1 ) ) or (
T1.u42_bp8objecttype = ? ) ) )
ORDER BY T1.u42_bp8objecttype DESC, T3.name , T2.u2e_documenttitle >
2006/02/25 01:04:00.250 0984 0230 InBindings0<{1B2F8C97-5D9F-4099-98C9-
6F4099A03359}, 66746, 1, 15>
```

The following is another example of a customized version of this query, which:

- . Assumes that the customer in question is not storing their attached documents in CE folders and
- . Uses the Email CE document subclass, along with the CaseManagementDocuments CE document subclass as an example to show how the CE SQL IsClass function can be used to

restrict the scope of a query to multiple specific CE document classes without searching all of the remaining document classes in CE.

This avoids the extra joins to both Folder and CustomObject, and returns only Documents.

```
select a.Bp8ObjectID as Id, a.Bp8ObjectType as ObjectType, a.Id as AttachID,  
d.DocumentTitle, d.VersionSeries, d.VersionStatus, d.MajorVersionNumber,  
d.MinorVersionNumber, d.MimeType, d.isReserved, d.IsVersioningEnabled from  
Bp8Attachment a left join Document d ON d.VersionSeries = a.Bp8VersionSeries  
where (IsClass(d, CaseManagementDocuments) or IsClass(d, Email)) and a.Bp8CaseID  
= ^1 and ^2 order by a.Bp8ObjectType desc, d.DocumentTitle asc
```

In the case of an actual BPF Case with Bp8CaseID 66746 and two attachments (one a Document and one another Case object), the ^1 and ^2 are being replaced to produce the following actual CE query:

```
select a.Bp8ObjectID as Id, a.Bp8ObjectType as ObjectType, a.Id as AttachID, d.DocumentTitle,
d.VersionSeries, d.VersionStatus, d.MajorVersionNumber, d.MinorVersionNumber, d.MimeType,
d.isReserved, d.IsVersioningEnabled from Bp8Attachment a left join Document d ON
d.VersionSeries = a.Bp8VersionSeries where (IsClass(d, CaseManagementDocuments) or
IsClass(d, Email)) and a.Bp8CaseID = 66746 and (((VersionStatus = 1 OR VersionStatus is null)
AND d.isCurrentVersion = True) OR a.Bp8ObjectType = 15) order by a.Bp8ObjectType desc,
d.DocumentTitle asc
```

According to the CE Catalog trace log, this CE SQL query is being converted to actual Microsoft T-SQL as follows by CE itself before being executed against the object store database:

```
2006/02/25 01:35:26.875 0bdc 0882 Text0<SELECT T1.u41_bp8objectid,
T1.u42_bp8objecttype, T1.object_id, T2.u2e_documenttitle, T2.version_series_id,
T2.version_status, T2.major_version_number, T2.minor_version_number,
T2.mime_type, T2.is_reserved, T2.versioning_enabled, T1.object_class_id,
T1.security_id, T2.object_class_id, T2.security_id

FROM   Generic T1

       left outer join (SELECT * FROM DocVersion D2 WHERE D2.home_id IS NULL) T2
on ( T2.version_series_id = T1.u47_bp8versionseries )

WHERE  T1.home_id IS NULL

       AND T1.object_class_id IN (?)

and ( ( ( T2.object_class_id = ? ) or ( T2.object_class_id = ? ) ) and (
T1.u3d_bp8caseid = ? ) and ( ( ( T2.version_status = ? ) or (
T2.version_status is null) ) and ( T2.is_current = 1 ) ) or (
T1.u42_bp8objecttype = ? ) ) )

ORDER BY T1.u42_bp8objecttype DESC, T2.u2e_documenttitle >

2006/02/25 01:35:26.875 0bdc 0882 InBindings0<{1B2F8C97-5D9F-4099-98C9-
6F4099A03359}, {54E7A9C7-68AC-42D6-B865-B155275B7D43}, {95D9616A-1800-4499-BE69-
8F6B64B82D91}, 66746, 1, 15>
```

NOTE Customers and project teams are responsible for developing and testing their own customized versions of this and the other CE queries configured for the Attachment tab and Viewer Documents List that can also be found in BPF Explorer under **Applications Settings > Web Application** as well, and could potentially be customized in similar ways. BPF does not guarantee any specific performance gains for any such customizations.

Case fields – Lookup fields

Lookups can be extremely expensive from a performance perspective. Internal lookups to the same CE/PE environment used by BPF typically take between 100 and 300 milliseconds. External custom lookups can obviously take much, much longer. Consider having a local copy of external data in the Bp8Metastore database, or doing without the lookup altogether, if possible.

Web Application logging

Configuring the <bpf_web_root>\WEB-INF\log4j.xml file as follows (see the sections highlighted in bold) is recommended as a way of reducing the level of logging as far as possible as a way of increasing performance accordingly:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```

<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

  <appender name="bp8" class="org.apache.log4j.DailyRollingFileAppender">
    <param name="File" value="{bp8.app_root}/WEB-INF/logs/bp8.log"/>
    <param name="DatePattern" value=".yyyy-MM-dd"/>
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d %p [%t] %c (%M:%L) - %m%n"/>
    </layout>
  </appender>

  <appender name="Profiling" class="org.apache.log4j.DailyRollingFileAppender">
    <param name="File" value="{bp8.app_root}/WEB-INF/logs/profiling.log"/>
    <param name="DatePattern" value=".yyyy-MM-dd"/>
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d %p [%t] %c (%M:%L) - %m%n"/>
    </layout>
  </appender>

  <appender name="external" class="org.apache.log4j.DailyRollingFileAppender">
    <param name="File" value="{bp8.app_root}/WEB-INF/logs/external.log"/>
    <param name="DatePattern" value=".yyyy-MM-dd"/>
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d %p [%t] %c (%M:%L) - %m%n"/>
    </layout>
  </appender>

  <logger name="com.filenet.wcm.api" additivity="false">
    <level value="error"/>
    <appender-ref ref="external"/>
  </logger>

  <logger name="Profiler" additivity="false">
    <level value="error"/>
    <appender-ref ref="Profiling"/>
  </logger>

  <logger name="com.filenet.bp8 " additivity="false">
    <level value="error"/>
    <appender-ref ref="bp8 "/>
  </logger>

```

```
<root>
  <level value="error"/>
  <appender-ref ref="external"/>
</root>

</log4j:configuration>
```

Avoid large dynamic pick lists

Dynamic pick lists are SQL statements that retrieve information from a backend database in real time. BPF does not cache the results of these SQL statements; they are executed each time the pick list is used. If a Case field is configured to use such a pick list, and then this field is configured as an Inbasket filter, a browse list column, and/or a regular Case field displayed on the Case tab for an Inbasket, then each time it is used/referenced in any of these contexts, a round-trip to the server will be required, during which the SQL statement will be executed and the results sent back to the client. For dynamic pick lists containing hundreds of items or more, this is obviously a prohibitively expensive process and alternative approaches should be considered. If the pick list is large because the design approach was to have a single table that is filtered at runtime to obtain a subset for a specific condition, consider creating separate smaller pick lists to support each requirement.

Avoid large numbers of Read From Workflow fields

Read From Workflow is used when the work object value may be more current than the case object value and you wish to read the work object value and synchronize it to the case object value upon Case open. The cost of having BPF update CE from PE when the values do not match is relatively high. Consider keeping the values synchronized by others means outside of BPF to avoid this overhead.

Annotation viewing

Currently there is a known significant performance issue in the FileNet Java Viewer when it checks for annotations. Until a fix pack is available, it is suggested that annotation checking (and display) be turned off, if possible. To implement this, the Java Viewer JSP file needs to be modified directly to disable annotations: `workplace\UI-INF\jsp\ui\WcmJavaViewer.jsp`. Contact CSS to request assistance in modifying this file.

Bulk Processing

The bulk processing function in BPF was designed to process only a small number of Cases at a time and there is a practical and physical limit as to how many objects can be processed at once. The recommendation is therefore to keep the number of items below 50 at a time and definitely no more than 100 at a time. In addition to performance considerations within BPF, there are CE limitations (length of the SQL query) that would not allow processing more than approximately 150 at a time.

Disable Inbasket(s) List on Search

Whenever a Case is opened from the Search Results list, BPF will display a list of all of the Inbaskets available to the current user's current Role or Security Profile in which that Case can be seen. If the Case's work object is no longer active in Workflow, a message to that effect is displayed instead.

In implementations where each Role or Security Profile is associated with a large number of

Inbaskets (more than 30 seems to be a general point where this issue is first noticed), users may begin to notice that it is taking a long time for BPF to open Cases from Search Results. This is because BPF executes the same SQL search against the PE database for each of the Inbaskets configured for the current user's current Role or Security Profile (just as if it were pulling up each Inbasket in the Browse List) and then displays the names of those Inbaskets for which count(*) is greater than 0. Once the number of Inbaskets involved here reaches 30 or so, the time it takes BPF to execute those 30 or more SQL queries can significantly slow down the opening of the Case.

For this reason, it is sometimes desirable to disable this feature entirely. Although future releases of BPF may provide a way to disable this feature through configuration, in BPF 3.6.0, it is necessary to make the following changes *manually* to the `openCaseQueryInbasket` function in the `Bp8InitMain.js` file (in **red** below):

```
function openCaseQueryInbasket() {
  if (inbaskets.length == 0) {
    showMessageScreen( localize( "QueryInbasketNotDefined", "Query Inbasket not
defined for this profile." ) );
    return;
  }
  var objWO;
  try {
    objWO = RemoteFunctions.getWorkObjectsDOM( initialCase );
  } catch (e) {return;}
  objWO=null; // Add this line to disable Inbasket(s) feature.
  var caseInbasketName = "";
  var firstCaseWob;
  /*-----*/
  // Add this line to disable Inbasket(s) feature.
  if(objWO==null){
    caseInbasketName="";
  }else{
  /*-----*/
  ...

  if(originCommandId == "9010"){
    caseInbasketName = caseInbasketName + " | " + localize(
"InsufficientPrivilegeToProcessCase", "Insufficient privilege to process this Case"
);
  }
  }
  } // Add this line to disable Inbasket(s) feature. DON'T FORGET THIS LINE!!!
  document.getElementById('tdRoleTitle').innerHTML = caseInbasketName;
  toolbarConfig( currentInbasketIndex );
  selectCase( initialCase, firstCaseWob );
}
```

Add the lines as indicated in red above, close any IE browser sessions currently pointing to BPF, clear the browser cache on each client machine, and then retry. The expected result is that you should not see any message indicating the Inbasket(s) list – and that your cases will open from the Search Results much more quickly.

Content Engine tuning

Query Options

If customers are experiencing high response times when browsing and/or searching an object store, adjusting the CE query options may prove beneficial. Two parameters may be tuned, and can be accessed as follows:

Content Engine tuning

1. Load Enterprise Manager.
2. Expand the Object Stores node.
3. Right click on your Object Store, and select Properties.
4. Select the Query tab. Under the *Query* tab, you can change the following parameters:
 - Enumeration Batch Size - The *Enumeration Batch Size* parameter refers to how many enumerated items the CE will fetch in one request. By default, the Enterprise Manager shows this value as zero. The actual batch size value used by the CE in this situation is 20. Tests performed in the test lab showed improved performance by increasing the parameters value to 50.
 - Default Query Batch Size - The *Default Query Batch Size* parameter refers to how many items the CE will fetch per roundtrip to the server. By default, the Enterprise Manager shows this value as zero. The actual batch size value used by the CE in this situation is 20. Tests performed in the test lab showed improved performance by increasing the parameters value to 50.

Disable trace logging

Consider turning off all trace logging through the Enterprise Manager, by navigating to the servers node and turning off trace logging for each one by right-clicking its Trace Logging node and selecting **Disable all Components**.

Index Content Engine Objects

Indexing certain table columns in the CE database can make a tremendous difference in terms of improving performance. The BPF Installation Guide recommends indexing the following BPF-specific properties:

Object	Index field name
Bp8Attachment	Bp8ObjectGUID
Bp8Attachment	Bp8Attachment_Bp8Case
Bp8AuditLogItem	Bp8AuditLogItem_Bp8Case
Bp8Case	Bp8CaseID

To accomplish this, do the following for each of the items previously listed using Enterprise Manager:

1. Navigate to **Enterprise Manager > {os display name} > Other Classes > Custom Object > Bp8Object**.
2. Right click on the appropriate object.
3. Select **Properties**.
4. Select **Property Definitions** tab.
5. Check the **Inherited Properties** check box.
6. Select the appropriate field name.
7. Click **Edit**.

8. Verify that the Indexed text box has value of single indexed. If not:
 - a. Click **Set/Remove** button.
 - b. Click **Set** radio button
 - c. Check **Single Indexed** checkbox.
 - d. Click **OK**.
9. Click **OK** to close properties page.

In addition, it has been found that putting database indexes on the object_class_id column of both the Generic and DocVersion tables in the CE object store (directly through the RDBMS) can provide a significant performance boost for BPF, even though object_class_id is not a BPF-specific property.

Process Engine tuning

In addition to the remarks found in the *IBM FileNet P8 Platform Performance Tuning Guide*, be sure to consult the FileNet P8 Platform documentation under the following topic:

[FileNet P8 Documentation](#) > [FileNet P8](#) > [Process Engine Administration](#) > [System administration tasks](#) > [Improving system performance](#)

Purge workflow event log

Consult the FileNet P8 Platform documentation under the following topic:

[FileNet P8 Documentation](#) > [FileNet P8 Administration](#) > [Process Engine Administration](#) > [Administrative Tools](#) > [vwlog](#) > [Purging data from the event logging table](#)

Disable unnecessary events

Certain event logging options cause slight performance degradation. The overall degradation is about five percent on the server, with no noticeable degradation on the PC workstation. You can, for example, enable or disable logging for a specific event category in an isolated region through the Process Configuration Console. For more information, see the Process Configuration Console online help. This can be done by un-checking the Work Item Messages you can do without from with Process Configuration Console under the Event Logging Options tab for the isolated region. Please consult the FileNet P8 Platform documentation for more details.

Create indices

Make sure to create indices on the queue fields defined as sortable columns in BPF Inbaskets.

Turn off VWRouter logging

In Process Task Manager, make sure that the logging is turned off for your router by setting the Debug Level to Off.

Monitor number of server processes

On the General tab of the Process Pooled Manager console, monitor the Server Processes in Use value and compare it with the Max Processes value configured there. If the value of Server Processes in Use is consistently near the Max Processes value, consider setting the Max Processes value higher.

Web Application container tuning

In general, be sure to consult the BEA documentation found at the following address for further details regarding the information found below: <http://edocs.bea.com/platform/docs81/>.

Internet Explorer settings

Cache settings

It is recommended that the IE cache setting is set to Automatic on user desktops to achieve best performance. To verify or correct the browser settings, open **Tools > Internet Options > General tab > Settings** and make sure the Check for Newer Versions of Stored Pages is set to Automatically. The BPF Web Application was optimized to take advantage of browser object caching in order to reduce network load and improve both server and client performance. Allowing for automated IE cache handling and having allocated at least 5 MB to cache is advantageous.

BEA WebLogic

Web container thread pool settings

The thread count value should be set to a value greater than or equal to maximum user load your application can have at a time. For example, if an application can have as many as 100 users connected and working at a time, then the minimum recommended value for max connections would be 100.

Turning off servlet reloading

Use a text editor to add the following child element with the following value to the container-descriptor element of the `weblogic.xml` file used by the BPF Web Application:

```
<container-descriptor>
...
<servlet-reload-check-secs>-1</servlet-reload-check-secs>
</container-descriptor>
```

Tuning the default execute queue threads

The value of the ThreadCount attribute of an ExecuteQueue element in the `config.xml` file equals the number of simultaneous operations that can be performed by applications that use the execute queue. As work enters an instance of WebLogic Server, it is placed in an execute queue. This work is then assigned to a thread that does the work on it. Threads consume resources, so handle this attribute with care—you can degrade performance by increasing the value unnecessarily.

By default, a new WebLogic Server instance is configured with a development mode execute queue, `weblogic.kernel.default` that contains 15 threads. In addition, the WebLogic server provides two other pre-configured queues:

- `weblogic.admin.HTTP` – Available only on administration servers, this queue is reserved for communicating with the administration console; you cannot reconfigure it.
- `weblogic.admin.RMI` – Both administration servers and managed servers have this queue; it is reserved for administrative traffic; you cannot reconfigure it.

Unless you configure additional execute queues, and assign applications to them, Web Applications and RMI objects use `weblogic.kernel.default`.

Modifying the default thread count

To modify the default execute queue thread count using the administration console:

5. Start the administration server if it is not already running.
6. Access the administration console for the domain.
7. Expand the Servers node in the left pane to display the servers configured in your domain.
8. Right-click the name of the server instance that contains the execute queue you want to configure, and then select **View Execute Queues** on the pop-up menu to display a table of execute queues that can be modified.

NOTE You can only modify the default execute queue for the server or a user-defined execute queue.

9. In the Name column, click directly on the default execute queue name to display the Configuration tab for modifying execute queues.
10. Locate the Thread Count value and increase or decrease it, as appropriate.
11. Click **Apply** to save your changes.
12. Reboot the selected server to enable the new execute queue settings.

Connection pool size

Maximum capacity

In the **JDBC Connection Services > JDBC > Connection Pool > Configuration > Connections** tab, you can specify the number of connections in the connection pool and details for each connection in the connection pool.

A connection pool contains a group of JDBC connections that are created when the connection pool is registered, usually when starting up WebLogic server. Your application borrows a connection from the connection pool, uses it, then returns it to the connection pool by closing it

This value should be set to a value greater than or equal to maximum user load your application can have at a time. For example if an application can have more up to 100 users connected and working at a time than minimum recommended value for max connections would be 100.

Initial capacity

The Initial Capacity setting specifies the minimum number of physical connections to maintain. Until this number is reached, the pool maintenance thread does not discard physical connections. This value should be set to a reasonable number depending upon customer's requirement.

Heap Size

Initial and maximum heap size

The preceding changes are made through the administration console and saved in BEA's `config.xml` file. By contrast, you need to make this change and the following changes by altering the actual script that is used to start up the BEA WebLogic server instance. Specifically, the `-Xmx` and `-Xms` flags passed to the `java.exe` need to be (re)configured.

For example:

```
-Xms<size>
```

-Xms sets the initial and minimum size of the heap. For this, we recommend that you set it to the same size as the maximum heap size; for example:

```
-java -Xgcprio:throughput -Xmx:64m -Xms:64m myClass
```

Please consult the online BEA documentation at <http://e-docs.bea.com/wls/docs81/perform/topten.html#1120355> for further details on the advantages of setting -Xmx and -Xms to the same value.

Initial heap size specifies the initial heap size available to the JVM code, in megabytes. Increasing the minimum heap size can improve startup. In general, increasing the size of the Java heap improves throughput until the heap no longer resides in physical memory. After the heap begins swapping to disk, Java performance suffers drastically.

The physical memory usage must be shared between the JVM and the other applications, such as, the database. For assurance, use a smaller heap, for example 64MB, on machines with less memory.

Try a maximum heap of 256MB on a smaller machine, that is, less than 1GB of physical memory, 512MB for systems with 2GB memory, and 1024MB for larger systems. The minimum heap size depends on the application. Setting the minimum heap size equal to the maximum avoids the onset of garbage collection activity. Subsequent garbage collection cycles are fewer, but have a longer duration that can affect performance.

As per the *IBM FileNet P8 Platform Performance Tuning Guide*, it is recommended that you adhere to the following BEA recommendations:

For clients running WLS 7.x, BEA recommends that 80 - 85% of the machines physical memory is allocated to the heap, as long as that value does not exceed 1 GB. Clients running WLS 8.x should follow the same rule, however they should not exceed a heap of 2 GB in size. In addition, BEA suggests that the minimum and maximum values for the heap be equal. For both versions of WLS, the suggestion above assumes that users are running a single instance of WLS.

The arguments listed below would be added to your startup script if you were to allocate a minimum and maximum size of 1 GB to your heap:

```
-Xms1024m -Xmx1024m
```

JIT compiler settings

Where possible, you should use a Just-in-Time (JIT) compiler when running the WebLogic server. JVMs with JITs include those from Sun Microsystems and Symantec. Good performance improvements have been noted from the use of JITs.

Please ensure that `-Djava.compiler=NONE` is NOT set in server startup script.

HTTP connection timeout

It is recommended to increase the HTTP connection timeout in from the default 5 seconds to 10 seconds or more seconds modifying `ConnectionIOTimeout` and `ConnectionKeepAliveTimeout` parameters in WebSphere administrative console. Changing these parameters is required to resolve timeout and xml parsing errors that might be observed on a slow or busy network connections. Follow the instructions in the link:

<http://publib.boulder.ibm.com/iserics/v1r1m0/websphere/ic2924/index.htm?info/rzaiz/50/admin/acshttpcp.htm>, specifically the following:

- **ConnectionIOTimeout** – Specifies the maximum number of seconds to wait when trying to read or process data during a request.

The default value is five seconds. This value determines how long the application server waits while receiving two subsequent data packets for the same HTTP request. For example, with the default `ConnectionIOTimeout` setting of five seconds, if an HTTP client sends two data packets spaced six seconds apart, the process times out, and the server throws a `java.io.InterruptedIOException` error. The server terminates the HTTP request, and the HTTP client must resubmit the request.

- **ConnectionKeepAliveTimeout** – Specifies the maximum number of seconds to wait for the next request on a keep-alive connection.

IBM WebSphere

Servlet caching

Configuring servlet caching

Enable servlet caching to save the output of servlets in memory using the dynamic cache service. Enabling the dynamic cache service improves application performance by caching the output of servlets, Web services, and WebSphere commands into memory

Before you begin

To enable servlet caching, you must complete enabling the dynamic cache service.

Steps for enabling servlet cache

1. Open the administrative console.
2. Click **Servers > Application Servers** in the console navigation tree.
3. Click a server.
4. Click **Web Container**.
5. Select **Enable servlet caching** under the Configuration tab.
6. Click **Apply** or **OK**.
7. Restart the WebSphere application server.

Steps for enabling dynamic cache service

You can enable dynamic cache service to improve response time. Use the administrative console to make configuration changes:

1. Select **Servers > Application servers > server_name > Container services > Dynamic cache service**.
2. Click on the Configuration tab.
3. Select the Enable Service at Server Startup option.

Web container thread pool settings

Requests are sent to the server through any of the HTTP transport channels or HTTP transports. A thread pool enables components of the server to reuse threads to eliminate the need to create new threads at run time. Creating new threads expends time and resources.

To view this administrative console page, click **Servers > Application Servers > server_name > Thread Pools > Thread Pool**. (You can reach this page through more than one navigational route.)

Maximum value should be set to a value greater than or equal to maximum user load your application can have at a time. For example if an application can have more than 100 users connected and working at a time than minimum recommended value for max connections would be 100.

Connection Pool size

JDBC max connections

Max connections specify the maximum number of physical connections that you can create in this pool. These are the physical connections to the backend resource. Once this number is reached, no new physical connections are created and the requester waits until a physical connection that is currently in use returns to the pool, or a `ConnectionWaitTimeoutException` is thrown.

This value should be set to a value greater than or equal to maximum user load your application can have at a time. For example if an application can have more than 100 users connected and working at a time than minimum recommended value for max connections would be 100.

JDBC min connections

Min connections specify the minimum number of physical connections to maintain. Until this number is reached, the pool maintenance thread does not discard physical connections. However, no attempt is made to bring the number of connections up to this number. If you set a value for Aged Timeout, the minimum is not maintained. All connections with an expired age are discarded. This value should be set to a reasonable number depending upon customer's requirement.

Heap size

Initial heap size specifies the initial heap size available to the JVM code, in megabytes. Increasing the minimum heap size can improve startup. For the WebSphere application server, the number of garbage collection occurrences is reduced and a 10% gain in performance is realized.

In general, increasing the size of the Java heap improves throughput until the heap no longer resides in physical memory. After the heap begins swapping to disk, Java performance suffers drastically.

The physical memory usage must be shared between the JVM and the other applications, such as, the database. For assurance, use a smaller heap, for example 64MB, on machines with less memory.

Try a maximum heap of 256MB on a smaller machine, that is, less than 1GB of physical memory, 512MB for systems with 2GB memory, and 1024MB for larger systems. The minimum heap size depends on the application. Setting the minimum heap size equal to the maximum avoids the onset of garbage collection activity. Subsequent garbage collection cycles are fewer, but have a longer duration that can affect performance.

As per the *IBM FileNet P8 Platform Performance Tuning Guide*, it is recommended that you adhere to the following IBM recommendations:

For Solaris customers running WS 5.0.x, it is recommended that 80 - 85% of the machines physical memory is allocated to the heap, as long as that value does not exceed 1 GB. Clients running WAS 5.1.x on Solaris should follow the same rule, however should not exceed a heap of 2 GB in size.

For Windows® customers running WebSphere, it is suggested that 80 - 85% of the machines physical memory be allocated to the heap regardless of the version of WS being used.

NOTE Both suggestions made above assume only a single instance of WS in running on the system.

The arguments listed below would be added to your startup script if you were to allocate a minimum and maximum size of 1 GB to your heap:

```
-Xms1024m -Xmx1024m
```

JIT compiler settings

The Just In Time (JIT) compiler can significantly affect performance. In all cases, run with JIT enabled, which is the default. To enable JIT:

1. Click **Servers > Application Servers > server_name**.
2. Click the **Process Definition > Java Virtual Machine** under Additional Properties.
3. Deselect the Disable JIT checkbox if selected.
4. Click **Apply** or **OK**.
5. Click **Save**.
6. Stop and restart the application server.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.



Program Number: 5724-R75

Printed in USA

SC19-2932-00

