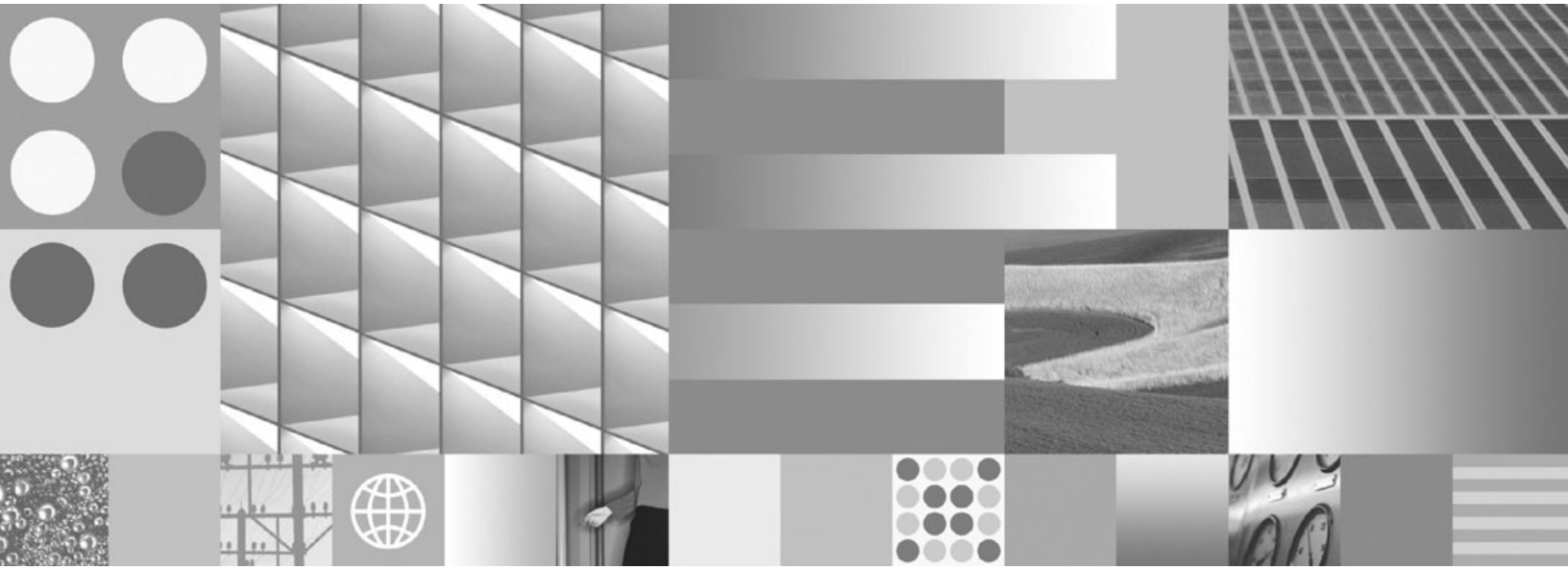
IBM FileNet P8 Content Engine

3.5

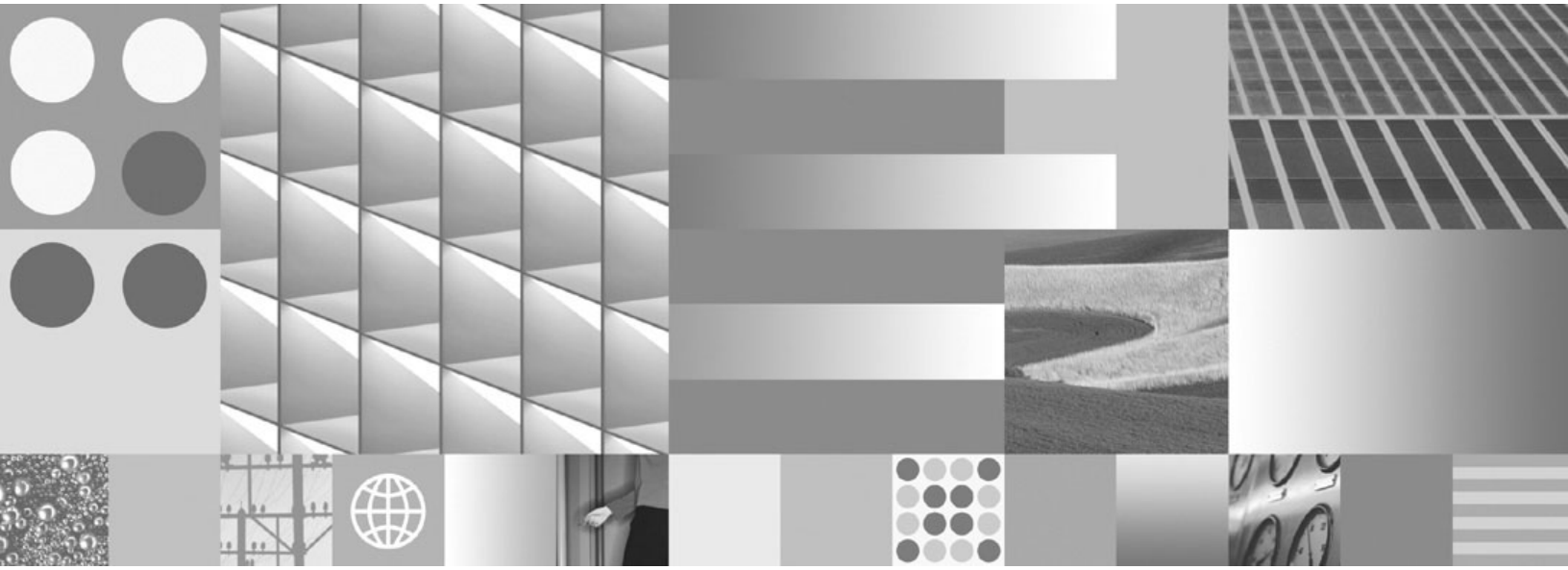


Query Performance Optimization Guidelines Technical Notice

GC31-5596-01

IBM FileNet P8 Content Engine

3.5



Query Performance Optimization Guidelines Technical Notice

GC31-5596-01

Note
Before using this information and the product it supports, read the information in "Notices" on page 15.

This edition applies to version 3.5 of IBM FileNet P8 Content Engine (product number 5724-R81) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Revision Log

Date	Revision
6/20/2006	Initial document.
5/7/2008	Added new guidelines: • Avoid LIKE searches with wildcard at the front • Avoid unnecessary row returns • Avoid unnecessary database locks
9/2/2009	Added new guidelines: • Avoid searches on multi-valued properties when using an OR condition • Avoid searches on the ContentSummary column (CBR) Modified guidelines: • Use the free-threading model (required)

Introduction

This document is a guide for optimizing the performance of your Content Java™ API or Content Engine COM API client SQL queries made against a IBM® FileNet® Content Engine server. This guide does not replace any application-specific performance studies or recommendations provided by IBM.

Audience

This document is for writers and readers of Content Engine SQL queries. This primarily means application developers, and also database administrators who are working with and assisting developers.

Accessing IBM FileNet Documentation

To access documentation for IBM FileNet products:

1. Navigate to the Product Documentation for FileNet P8 Platform support page:

   http://www.ibm.com/support/docview.wss?rs=0&dc=DA400&q1=FileNet+P8&uid=swg27010422&loc=en_US&cs=utf-8&cc=us&lang=en

2. Select a PDF or a Doc Link, whichever is appropriate.

References

For general database tuning information, see *FileNet P8 Platform 3.5.x Performance Tuning Technical Notice*. To download this guide from the IBM support page, see Accessing IBM FileNet Documentation.

For information on Content Engine SQL syntax, see FileNet P8 Documentation **>** Developer Help > Developer Roadmap > Reference > SQL Syntax Descriptions.

For general information on queries using the Content Java API or Content COM API, see:

- FileNet P8 Documentation **>** Developer Help > Content Java API > Searching
- FileNet P8 Documentation **>** Developer Help > Content Engine COM API > COM API Guide > Using the Content Engine Database Engine > Querying Information About Document Objects

NOTE For guideline-specific references, see the appropriate topic under Guidelines.

Using Query Builder

Use the FileNet Enterprise Manager's Query Builder tool to visually construct your query or to quickly validate that your query works as intended. To run Query Builder, select an object store and then select Action > Search.

Be sure to include the object reference “this” in the SELECT clause when attempting to execute any query examples from this technical notice in Query Builder > View > SQL View. For instance, enter the example query

```
SELECT d.Id FROM Document d WHERE d.DocumentTitle = 'MyDoc'
```

into Query Builder in this form:

```
SELECT d.this, d.Id FROM Document d WHERE d.DocumentTitle = 'MyDoc'
```

For more information on using the Query Builder, see FileNet P8 Documentation > Developer Help > Content Engine COM API > COM API Guide > Using the Content Engine Database Engine > Querying Information About Document Objects > Using the Content Engine Query Builder.

Creating Indexes

As indicated by several of the guidelines, your query's performance might depend on the existence of database indexes for various Content Engine object properties. In most cases, use the FileNet Enterprise Manager as the most convenient means to create these indexes; see FileNet P8 Documentation > FileNet P8 Administration > Content Engine Administration > Classes > How to... > View/modify class property definitions > Set/remove indexing. In some cases you might need to use the tools provided by the database engine (Oracle, SQL Server, etc.) either to verify that the expected index exists or to create the index (when unable to do so using Enterprise Manager), and consequently need to know the table column that corresponds to the object property in question.

For class-to-table mappings, see FileNet P8 Documentation > Developer Help > Content Engine COM API > COM API Guide > Using the Content Engine Database Engine > Using Content Engine Database Tables > Classes and Database Tables. Note that the table column name, when not identical to the object property name, will have this format: xxx_<object property name>, where "xxx" is some generated prefix. For example, the object property Document.DocumentTitle might map to table column DocVersion.u2e_documenttitle.

Guidelines

Limit rows returned

Limit the number of rows returned from your query to a usable number, either by tailoring your query's row selection criteria to achieve that result, or by arbitrarily deciding on that number in advance by setting the Microsoft® ActiveX Data Object (ADO) MaxRecords property.

When using the Content Java API, set the MaxRecords property indirectly via the passed XML. For an example, see FileNet P8 Documentation > Developer Help > Content Java API > Searching > Working with Search-related Objects > Performing an Ad hoc Search.

When using the Content COM API, set the property directly as shown in this example:

```
Set ADOCnn = ObjStore.GetADOConnection

Set RecSet = CreateObject("ADODB.Recordset")

RecSet.MaxRecords = 20

Call RecSet.Open(Query, ADOCnn, adOpenStatic, adLockReadOnly)
```

Avoid non-indexed ordering and searching

Avoid referencing any non-indexed column in a JOIN, WHERE, or ORDER BY clause. For example, optimize the query below by creating an index on Document.DocumentTitle (database column DocVersion.xxx_documenttitle):

```
SELECT d.Id FROM Document d WHERE d.DocumentTitle = 'MyDoc'
```

NOTE The Document class maps to the DocVersion table. See Creating Indexes.

For indexing discussions, see the *FileNet P8 Platform 3.5.x Performance Tuning Technical Notice*.

Avoid LIKE searches with wildcard at the front

Avoid WHERE clauses with the LIKE operator when the searched-for column value does not permit the effective use of indexes. For example:

```
SELECT d.Id FROM Document d WHERE d.DocumentTitle LIKE '%abc'
```

In this example, even if an index exists for the DocumentTitle column, the query cannot use it due to the leading wildcard character ("%") in the searched-for column value. Code your query to return a relatively small number of rows by placing a sufficient number of characters before the wildcard. In search dialogs, allow end users to perform "Starting with" searches (such as for "abc%") but not "Contains" searches (such as for "%abc%").

Queries that do not make effective use of indexes can potentially cause the database to lock the table. Other users can thus be blocked from updating the table for the duration of the query, causing checkins to time out and other such problems.

Avoid non-function-indexed case-insensitive comparisons (Oracle / DB2)

Avoid any column value comparison resulting from a JOIN, WHERE, or ORDER BY clause for any non-indexed column, or for any column belonging to an index that does not directly or indirectly use the LOWER function. Follow this guideline in these circumstances: you are working with an object store that uses Oracle or DB2® as the database engine, you have forced case-insensitive searches by setting the ForceCaseInsensitiveSearch property to true, and the Oracle or DB2 database has not been natively configured for case-insensitivity (and so setting ForceCaseInsensitiveSearch to true has a practical effect).

For DB2, generate a column by applying the LOWER function to a pre-existing column, and then create an index on the generated column. For Oracle, create a function-based index. For example, the index LOWER(DocVersion.xxx_documenttitle) makes the following queries more efficient:

```
SELECT d.Id FROM Document d WHERE d.DocumentTitle = 'MyDoc'

SELECT d.Id, d.DocumentTitle, d.Creator FROM Document d ORDER BY d.DocumentTitle
```

NOTE The Document class maps to the DocVersion table. See Creating Indexes.

For additional information on performance issues associated with case-insensitive queries, see the *FileNet P8 Platform 3.5.x Performance Tuning Technical Notice*.

For additional information on case-insensitive queries or the ForceCaseInsensitiveSearch property, see:

- FileNet P8 Documentation **>** FileNet P8 Administration > Content Engine Administration > Search and bulk operations > How to > Force case insensitive searches
- FileNet P8 Documentation > Developer Help > Content Java API > Searching > Working with Search-related Objects > Performing an Ad hoc Search
- FileNet P8 Documentation **>** Developer Help > Content Engine COM API > COM API Reference > Properties > ForceCaseInsensitiveSearch Property

Avoid unnecessary row returns

When using ADO, use the server cursor (the default cursor type). The server cursor minimizes the amount of data transmitted over the network, whereas the client cursor causes all searched-for rows to be returned to the client (versus just the initial batch, or those batches of rows actually read by the application).

Avoid unnecessary object type searches

Avoid unnecessary searching for subclass types by adding the EXCLUDESUBCLASSES operator to your query. You need to explicitly add this operator because a query has an implicit INCLUDESUBCLASSES operator by default.

For example, as a result of the implicit INCLUDESUBCLASSES, the following query might return objects belonging to a Document subclass, in addition to those belonging to the Document class:

```
SELECT d.Id FROM Document d WHERE DocumentTitle = 'MyDoc'
```

Presuming only objects from the Document class are needed, instead of writing the query as

```
SELECT d.Id FROM Document d WHERE DocumentTitle = 'MyDoc' AND ISCLASS(d, Document)
```

use the EXCLUDESUBCLASSES operator in this manner:

```
SELECT d.Id FROM Document d WITH EXCLUDESUBCLASSES WHERE DocumentTitle = 'MyDoc'
```

For a discussion of the EXCLUDESUBCLASSES operator, see FileNet P8 Documentation > Developer Help > Developer Roadmap > Reference > SQL Syntax Descriptions > Include/Exclude Subclasses Function Description.

Avoid unnecessary column returns

Avoid returning all of the columns in a table when only some are needed. For example, instead of

```
SELECT d.* FROM Document d WHERE d.DocumentTitle = 'MyDoc'
```

specify the needed columns, as in this example:

```
SELECT d.Id FROM Document d WHERE d.DocumentTitle = 'MyDoc'
```


This minimizes the amount of data transmitted over the network. Also, when all of the columns in the SELECT clause are indexed, the data for the query might be retrieved directly from the index, thereby making the physical row lookup unnecessary.

Avoid unnecessary database locks

When using the Content COM API and ADO to perform searches, close the ADO recordset once the desired rows have been retrieved and before performing other tasks. This minimizes the amount of time that other users might be affected by database locks held by the query. If you are going to perform updates based on the returned rows, first retrieve the rows, store the needed row identifiers (in an array or file, for example), close the recordset, and then perform the updates using the stored row identifiers. Recordsets are automatically closed for you when searching through the Java API.

Use the free-threading model (required)

Use the free-threading model for Microsoft ActiveX Data Objects (ADO) on the Content Engine server. Using this model can improve the performance of your queries, including those executed via the Content Java API. In particular, you must use this model to avoid problems when running multiple queries simultaneously.

For details on changing the ADO threading model, see FileNet P8 Documentation > Developer Help > Content Engine COM API > COM API Guide > Using the Content Engine Database Engine > Querying Information About Document Objects.

For a discussion of the Content Java API Listener, and its implementation using COM objects, see FileNet P8 Documentation > Developer Help > Content Java API > Overview.

Tune query batch size parameters

Optimize query response time by tuning the Default Query Batch Size and Enumeration Batch Size parameters. These parameters control the size of the data chunks returned by the Content Engine server. For more information, see the *FileNet P8 Platform 3.5.x Performance Tuning Technical Notice*.

Avoid searches on multi-valued properties when using an OR condition

Queries that contain two or more multi-valued property conditions in an OR clause can result in query timeouts or unacceptable query performance because it can be difficult for the database engine to create an optimal query plan. To avoid this situation, issue each of the multi-valued property query conditions as a separate search.

Avoid complex table linkages

Avoid referencing three or more tables in a query. More tables might degrade query performance, and complicate performance tuning efforts.

Avoid unnecessary result row ordering

Avoid an unnecessary ORDER BY clause. Explicit row ordering might be unnecessary when you can rely on your query returning its results in a particular order as a side effect of an indexed search. For example, the ORDER BY clause in the query below might be unnecessary since a composite index exists for the Container table on (parent_container_id, name).

```
SELECT f.FolderName FROM Folder f WHERE f.Parent = Object('/sub1/sub1a') AND
f.IsHiddenContainer = false ORDER BY f.FolderName
```

NOTE The Folder class maps to the Container table.

Avoid subqueries (Oracle)

Avoid using subqueries and operators that indirectly generate subqueries whenever you are querying an object store that uses Oracle as the database engine. Potential subquery-related issues exist with the Oracle optimizer.

Specifically, use an INNER JOIN instead of a subquery (although see the Avoid complex table linkages guideline). For example, re-write the potentially slow query

```
SELECT d.Id FROM Document d WHERE d.Id IN (SELECT b.Bp8ObjectGuid FROM Bp8Attachment b)
```

in this functionally equivalent form:

```
SELECT d.Id FROM Document d INNER JOIN Bp8Attachment b ON d.Id = b.Bp8ObjectGuid
```

NOTE The FileNet Business Process Framework product uses the Bp8Attachment table.

Also, because the Content Engine uses a subquery to implement the INFOLDER operator, avoid using that operator. For example, re-write the query

```
SELECT f.FolderName FROM Folder f WHERE f.this INFOLDER '/sub1/sub1a' AND
f.IsHiddenContainer = false
```

in this manner:

```
SELECT f.FolderName FROM Folder f WHERE f.Parent = OBJECT('/sub1/sub1a') AND
f.IsHiddenContainer = false
```

Also, re-write

```
SELECT d.id FROM Document d WHERE d.This INFOLDER '/sub1/sub1a'
```

in this more efficient form:

```
SELECT d.Id FROM Document d INNER JOIN ReferentialContainmentRelationship r ON d.This =
r.Head WHERE r.Tail = OBJECT('/sub1/sub1a')
```

For a discussion of the INFOLDER operator, see FileNet P8 Documentation > Developer Help > Developer Roadmap > Reference > SQL Syntax Description > Folder Operators Description.

Avoid searches on the ContentSummary column (CBR)

Avoid referencing the ContentSummary column in a content-based retrieval (CBR) search, especially when using a Content Engine release before release 3.5.2-014. The additional retrieval of this text column slows the execution of the search.

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 (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.
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 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 contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

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.

Microsoft is a trademark of Microsoft Corporation in the United States, other countries, or both.

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

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

U.S. Patents Disclosure

This product incorporates technology covered by one or more of the following patents: U.S. Patent Numbers: 6,094,505; 5,768,416; 5,625,465; 5,369,508; 5,258,855.

Program Number: 5724-R81

Printed in USA

GC31-5596-01