IBM® Security Identity Manager

Version 6.0

API Performance Tuning Guide

SC147-396-00

Note: Before using this information and the product it supports, read the information in “Notices”.

Edition notice

Note: This edition applies to version 6.0 of IBM Security Identity Manager (product number 5724-C34) and to all subsequent releases and modifications until otherwise indicated in new editions.

Table of Contents

About this publication

This guide identifies best practices for application developers when using the IBM® Security Identity Manager application program interface (API).

Use this guide if you are responsible for using the IBM Security Identity Manager system APIs. The following competencies are recommended:

- Familiarity with basic database and directory design principles.
- General knowledge of the Java programming language.

Contributors

- Nnaemeka Emejulu
- Tommy Nguyen
- Daniel Nieman
- Karthik Ramamoorthy
- Anil Levi

Access to publications and terminology

This section provides:

- A list of publications in the IBM Security Identity Manager library.
- Links to "Online publications."
- A link to the "IBM Terminology website."

IBM Security Identity Manager library

The following documents are available in the IBM Security Identity Manager library:

- *IBM Security Identity Manager Quick Start Guide*, CF3L2ML
- *IBM Security Identity Manager Product Overview Guide*, GC14-7692
- *IBM Security Identity Manager Scenarios Guide*, SC14-7693
- *IBM Security Identity Manager Planning Guide*, GC14-7694
- *IBM Security Identity Manager Installation Guide*, GC14-7695
- *IBM Security Identity Manager Configuration Guide*, SC14-7696
- *IBM Security Identity Manager Security Guide*, SC14-7699
- *IBM Security Identity Manager Administration Guide*, SC14-7701
- *IBM Security Identity Manager Troubleshooting Guide*, GC14-7702
- *IBM Security Identity Manager Error Message Reference*, GC14-7393
- *IBM Security Identity Manager Reference Guide*, SC14-7394
- *IBM Security Identity Manager Database and Directory Server Schema Reference*, SC14-7395

- *IBM Security Identity Manager Glossary*, SC14-7397

Online publications

IBM posts product publications when the product is released and when the publications are updated at the following locations:

IBM Security Identity Manager Information Center
The http://pic.dhe.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.isim.doc_6.0/ic-homepage.htm site displays the information center welcome page for this product.

IBM Security Information Center
The http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp site displays an alphabetical list of and general information about all IBM Security product documentation.

IBM Publications Center
The http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss site offers customized search functions to help you find all the IBM publications you need.

IBM Terminology website

The IBM Terminology website consolidates terminology for product libraries in one location. You can access the Terminology website at http://www.ibm.com/software/globalization/terminology.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For additional information, see the topic "Accessibility features for IBM Security Identity Manager" in the *IBM Security Identity Manager Reference Guide*.

Technical training

For technical training information, see the following IBM Education website at http://www.ibm.com/software/tivoli/education.

Support information

IBM Support provides assistance with code-related problems and routine, short duration installation or usage questions. You can directly access the IBM Software Support site at http://www.ibm.com/software/support/probsub.html.

IBM Security Identity Manager Troubleshooting Guide provides details about:

- What information to collect before contacting IBM Support.
- The various methods for contacting IBM Support.
- How to use IBM Support Assistant.
- Instructions and problem-determination resources to isolate and fix the problem yourself.

Note: The Community and Support tab on the product information center can provide additional support resources.

Chapter 1. Introduction

The IBM Security Identity Manager product provides an API that customers often use to integrate IBM Security Identity Manager functionality into existing processes or customer interfaces. As IBM Security Identity Manager deployments grow in size, scalability and performance become more crucial for customer satisfaction.

The API allows great flexibility for accessing and manipulating information. This flexibility can often result in developers using the API in ways that yield poor performance or in ways that do not scale. This document describes some best practices on how to use the API while maximizing performance and scalability.

All tips in this guide are applicable to IBM Security Identity Manager servers at V5.1 and later, unless otherwise specified.

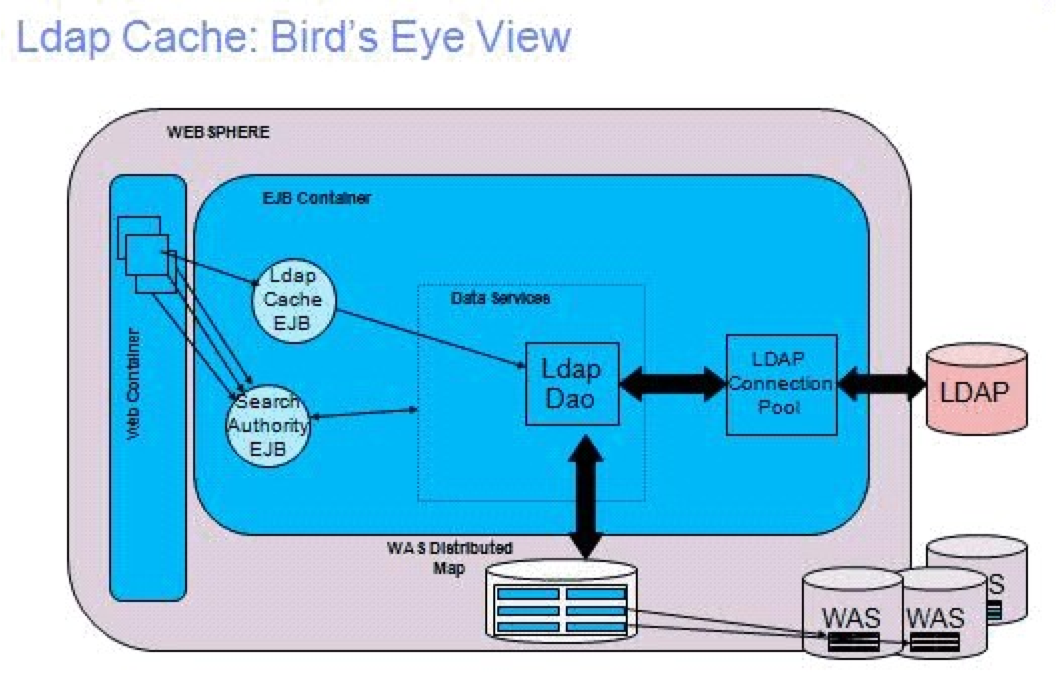
This is a supplement, not a replacement, to the IBM Security Identity Manager API documentation shipped with the product.

This document is a working document. As more information is gathered, settings will be added, removed or changed in future editions. Check the IBM website for the most recent version at: http://www-1.ibm.com/support/docview.wss?uid=swg27036204.

Chapter 2. Local and remote functions

The IBM Security Identity Manger provides two classes of API functions: local and remote. Local API functions are those designed to be used from within the IBM Security Identity Manager product, such as within a custom workflow extension or a javascript extension. Remote API functions are designed to be called from an external process, such as a custom interface or abstraction layer.

Invoking non-remoteable ITIM APIs, such as ProvisioningPolicyAnalysis or other Data Services APIs, outside of the ITIM classloader may result in ClassCastExceptions related to the LDAP Cache. This is due to the LDAP cache model used by ITIM which utilizes WebSphere’s Distributed Object cache. When invoked remotely, the remote application creates a new LdapCacheKey locally which is pushed to the WAS Distributed Map. Subsequent ITIM server operations will retrieve this new LdapCacheKey and could throw a class cast exception upon finding the retrieved key does not match the expected key local to the ITIM server.



Ldap Cache: Bird’s Eye View

Remoteable APIs are only those packages under com.ibm.itim.apps.* (such as com.ibm.itim.apps.policy.ProvisioningPolicyManabger).
All other documented APIs should be considered non-remoteable and should not be invoked remotely.

Chapter 3. Tips for remote APIs

The following tips apply only to remote API calls.

Avoid getResults() for large result sets

When using SearchResultsMO (or SearchMO.execute()) for large result sets, avoid using getResults(). Instead, get the results one page at a time using getPage() with a reasonable pageSize on the SearchMO object.

Impact

Returning the results from the SearchMO object in batches will reduce the amount of memory required in the IBM Security Identity Manager JVM, thereby preventing an out of memory (OOM) error. Fetching the data one page at a time ensures that manageable chunks of data are transmitted from the server to the client.

Increasing the pageSize in the SearchMO object will improve throughput at the cost of more memory usage on the server side and larger serialized objects.

Example

```
SearchMO searchMO = new SearchMO(platform, subject);
searchMO.setPageSize(100);

SearchResultsMO res = SearchMO.execute();
int pageNumber = 0;
Collection localResults = new ArrayList();
Collection page = res.getPage(++pageNumber);
while(page != null && page.size() == 100) {
  local.addAll(page);
  page = res.getPage(++pageNumber);
}

if(page != null && pageNumber == 1) {
  local.addAll(page);
}
```

Use getPageCount(int) where possible

The SearchResultsMO.getPageCount() function is commonly used to return the total number of pages in the result set. When possible, callers should use the getPageCount(int lookaheadLimit) instead.

Impact

The getPageCount() function requires loading the entire result set, which takes time and server memory.

Often, particularly for user interfaces, it is sufficient to see if there are more than a specific number of pages rather than needing the total number. In this situation, it is preferable to call the getPageCount(int lookaheadLimit) instead. This allows the server to load only the subset of data requested.

Example

```
SearchMO searchMO = new SearchMO(platform, subject);
searchMO.setPageSize(100);

SearchResultsMO res = SearchMO.execute();

// in our UI, we want to show 2 pages and only load subsequent pages
// if requested by the user. To that end we’ll request 3 pages, show
// 2 of them, and only show the ‘next’ button if getPageCount(3) returns 3
int pageCount = res.getPageCount(3);
```

Use ‘`null`’ as argument for a schedule to start immediately

Several `AccountMO`, `PersonMO`, and `RoleMO` functions accept a `java.util.Date` parameter used to schedule the action for some time in the future. If you want the operation to start immediately, use `null` as the argument.

Impact

Passing in a new `Date()` value has two negative impacts:

- Specifying a date instead of `null` results in a message for future activity being added to the `SCHEDULED_MESSAGE` table, which creates unnecessary work for IBM Security Identity Manager. Also, this might result in lock contention for busy systems. These impacts are important to consider.
- If the API call is done on a remote system, the current system’s date may not match the server’s date due to being out of sync or timezone differences, resulting in undesired behavior.

Example

```
// to have a person update happen immediately, use null as the second argument
personMO.update(person, null)

// ditto for account suspends
accountMO.suspend(null);

// and removing a member from a role
roleMO.removeMember(person, null);
```

Chapter 4. Tips for local APIs

The following tips apply only to local API calls.

Avoid `toCollection()` and `size()` for large result sets

Avoid calling `toCollection()` or `size()` on a `SearchResult`. Instead, use an iterator to process each result entity.

Impact

Calling `toCollection()` on a `SearchResult` will cause the entire result set to reside in memory as a collection. Similarly, calling `size()` on a `SearchResult` will force traversal of the entire list.

Instead, it is a better practice to use an iterator to iterate through the results.

Example

```
SearchResults sr = null;
try {
  PersonSearch perSearch = new PersonSearch();
  sr = perSearch.searchByFilter()
  SearchResultsIterator e = sr.iterator();
  while(e.hasNext()) {
    PersonEntity entity = (PersonEntity) e.next();
    // code to process each entity would go here
    e.remove(); // remove it after being used to free up memory
  }
} finally {
  if (sr != null)
    sr.close();
}
```

Chapter 5. Tips for local and remote APIs

The following tips apply to both local and remote API calls.

Avoid using `(objectclass=*)` where possible

To return all objects from a search, use an empty filter instead of using `(objectclass=*)`.

Impact

Filters used with `AccountSearch`, `SearchMO`, and related functions are often modified before being sent to the directory server. Specifying `(objectclass=*)` for a filter when all results are requested might result in a non-optimal filter being sent to the directory server. Instead, pass an empty string to the API.

Example - Remote API

```
/* Code to set max size, when there –
  A. Is a guarantee that the filter is bound to match a single unique entry
  B. Run an existence search only
*/
SearchMO searchMO = new SearchMO(platform, subject);
// Set sizeLimit to 2 for case A, to ensure no duplicate entities
// Set sizeLimit to 1 for case B, to check for consistence
searchMO.setMaxSize(2);

/* Set return attributes */
String [] retAttrs = new String[2];
retAttrs[0] = "cn";
retAttrs[1] = "title";
searchMO.setAttributes(retAttrs);
```

Example - Local API

```
/* Set search result limit */
SearchParameters params = new SearchParameters();
params.setSizeLimit(2);

/* Set return attributes */
ArrayList retAttrs = new ArrayList();
retAttrs.add("cn");
retAttrs.add("title");
params.setAttributes(retAttrs);
```

Specify return attributes and size limit for searches

When searching for an entity, explicitly specifying the desired attributes, specifying a limit on the number of entities returned, or both will improve performance.

Impact

By specifying a list of desired attributes, the directory server can minimize the amount of data returned by decreasing the time required to return the results. Specifying an attribute list can also reduce the memory overhead of the resulting data set.

Setting a limit on the resulting result set can provide information to the directory server to allow it to optimize the search for the data, as well as decreasing the memory overhead of undesired results.

Specifying a limit of 1 or 2 is useful for calls designed to check for the existence or uniqueness of an entity.

Example - Remote API

```
/* Code to set max size, when there –
  A. Is a guarantee that the filter is bound to match a single unique entry
  B. Run an existence search only
*/
SearchMO searchMO = new SearchMO(platform, subject);
// Set sizeLimit to 2 for case A, to ensure no duplicate entities
// Set sizeLimit to 1 for case B, to check for consistence
searchMO.setMaxSize(2);

/* Set return attributes */
String [] retAttrs = new String[2];
retAttrs[0] = "cn";
retAttrs[1] = "title";
searchMO.setAttributes(retAttrs);
```

Example - Local API

```
/* Set search result limit */
SearchParameters params = new SearchParameters();
params.setSizeLimit(2);

/* Set return attributes */
ArrayList retAttrs = new ArrayList();
retAttrs.add("cn");
retAttrs.add("title");
params.setAttributes(retAttrs);
```

Optimize `SearchMO` context

When using the `SearchMO` object, directory server performance can be improved by specifying an optimal logical context. The logical context, set with the `SearchMO.setContext()` function, specifies the base within the Organizational Tree. Whenever possible, specify the lowest organizational context that will return the desired result set rather than using the tenant context.

Impact

When a tenant-level logical context is specified and the recycle bin is enabled, the resulting LDAP filter to the directory server will include the filter `(!(erIsDeleted=Y))` to remove any entries contained in the recycle bin. This filter results in poor performance of the directory server. By specifying an organizational-level logical context, this filter is not required and can yield better directory server performance.

In addition, by specifying an organizational-level context, the directory server can evaluate a smaller number of possible matches.

The recycle bin can be disabled in IBM Security Identity Manager V4.5.1 IF 83 or higher and V4.6 IF 54 or higher. It is disabled by default in V5.0. Consult the *Performance Tuning Guide* for more information on disabling the recycle bin.

Explanation

When a person search is done with a filter and a tenant logical context, the resulting call to the directory server will look something like:

Base1 – <*tenantDN*>

Filter1 – (&(<*user_specified_filter*>)(!(erisDeleted=Y)))

When the same filter is used with the organizational context, the resulting call is:

Base2 – ou=people, <*organizationDN*>

Filter2 – (<*user_specified_filter*>)

Filter2 will yield faster results than Filter1. Also, using the organization DN as the base narrows the search to a smaller set within the directory server.

Example - Remote API

```
// Code to get compound DN object – do this one time
String primaryOrgName = "ACME"; // Primary Organization Name, usually same as tenantId
ContainerManager cm = new ContainerManager(platform, subject);
Collection containers = cm.getContainers("organization", primaryOrgName, null);
OrganizationalContainerMO orgMO =
  (OrganizationalContainerMO) containers.iterator().next();
// Create a compound DN with both the Tenant DN and the Org Unit DN
CompoundDN cdn = new CompoundDN(cm.getRoot().getDistinguishedName());
cdn.append(orgMo.getDistinguishedName());

// Code to set the context in SearchMO using the compound DN – do this as needed
searchMO searchMO = new SearchMO(platform, subject);
searchMO.setContext(cdn); // Prevents ‘((erIsDeleted=y))’ condition
searchMO.setScope(SearchMO.SUBTREE_SCOPE); // Ensures subtree search
```

Example - Local API

```
CompounDN searchCtxt = new CompounDN(tenantDN);
searchCtxt.append(<ORG_DN>);
SearchParameters params = new SearchParameters();
params.setScope(SearchParameters.SUBTREE_SCOPE);
SearchResults searchRes =
  AccountSearch().searchByFilter(searchCtxt, profileName, filter, params);
```

Close `SearchResults` and `SearchResultsMO`

When processing search results using `SearchResults` or `SearchResultsMO`, always close them when the results are no longer needed. For example, do this after the last page is accessed from a paged search or in a `finally` block.

Impact

Both functions internally cache the LDAP connection to the directory server. Failing to close the result set can lead to exhausting the available LDAP connections in addition to holding onto memory that would otherwise be freed.

Example - Remote API

```
SearchMO searchMO = new SearchMO(platform, subject);
// code to set up search goes here
SearchResultsMO searchRes = null;
try {
  searchRes = searchMO.execute();
  // code to process search results goes here
} finally {
  // close SearchResultsMO
  if(searchRes != null) {
  try {
    searchRes.close();
  } catch(Exception e) {}
}
```

Example - Local API

```
SearchResults searchRes = null;
try {
  searchRes =
    new AccountSearch().searchByFilter(searchCtxt, profileName, filter, params);
  // code to process search results goes here
} finally {
  // close SearchResults
  if(searchRes != null) {
  try {
    searchRes.close();
  } catch(Exception e) {}
}
```

Avoid using `SearchMO.setSortAttribute()` when possible

The `SearchMO.setSortAttribute()` method can be used to sort the search results by a specific attribute. Sorting the results places additional load on the LDAP server if server-side sorting is enabled or on the IBM Security Identity Manager server if server-side sorting is disabled and should not be used if sorted results are not required.

Impact

If server-side sorting is enabled in the `enRole.properties` file, sorting even small result sets can result in poor LDAP server performance. If server-side sorting is disabled, sorting large result sets can be CPU and memory intensive on the IBM Security Identity Manager server.

Chapter 6. Other resources

You will find the following resources useful for further tuning of IBM Security Identity Manager:

- *IBM Security Identity Manager 5.x Performance Tuning Guide*: http://www-1.ibm.com/support/docview.wss?uid=swg27011444
- *IBM Security Identity Manager 6.0 Performance Tuning Guide*: http://www-01.ibm.com/support/docview.wss?uid=swg27036205

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features contained 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 might have patents or pending patent applications that cover 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.

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 might 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 to enable: (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 might 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 might vary significantly. Some measurements might 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 might have been estimated through extrapolation. Actual results might 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 the future direction or intent of IBM 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 that conform 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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp.

2004, 2013. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations might not appear.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both: http://www.ibm.com/legal/copytrade.shtml

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

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