Sametime
Version 9.0

Sametime 9.0
Software Development Kit
Organization Tree Service Guide

Edition Notice

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

This edition applies to version 9.0 of IBM Sametime (program number 5725-M36) and to all subsequent releases and modifications until otherwise indicated in new editions.

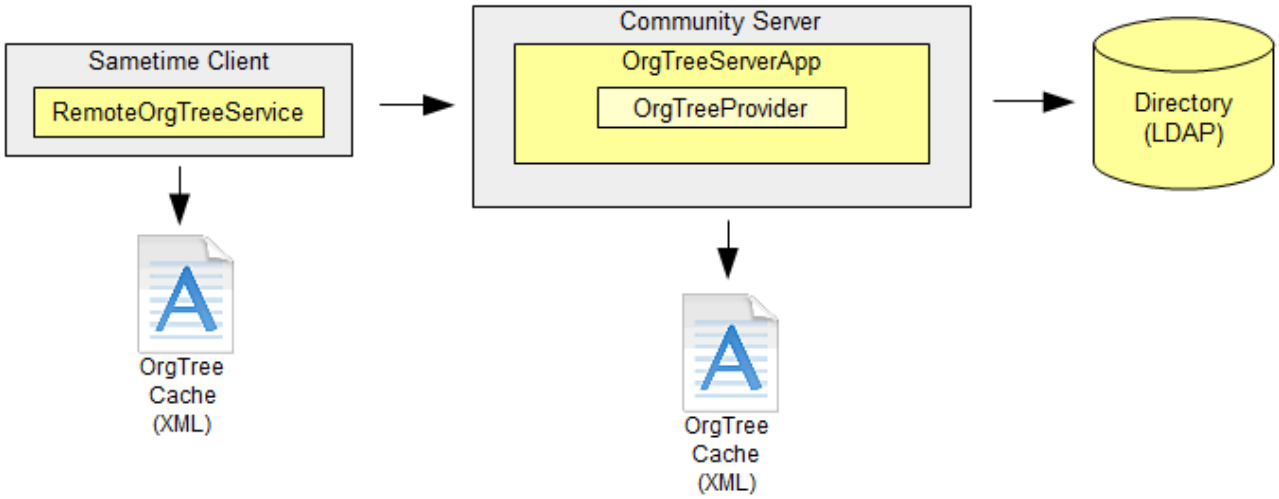
Contents

Chapter 1. Overview

The organization tree service is a community server application (SA) that provides a hierarchical view of the company directory to the Sametime client. The service is configurable through a set of configuration properties and may also be extended to suit the needs of different directories and environments. This document describes the structure, deployment, and extensibility of the organization tree service.

Chapter 2. Organization tree service architecture

The organization tree service is deployed as a remote SA as shown in the figure below.



The client uses a RemoteOrgTreeService which communicates with the OrgTreeServerApp using VP through the community server. The OrgTreeServerApp in turn delegates the client requests to the OrgTreeProvider, which queries the directory. The results are then sent back to the client by the OrgTreeServerApp and cached by both the OrgTreeServerApp and client.

To minimize load on the directory, both the client and the remote SA maintain a persistent, configurable cache. Before making a remote call to the OrgTreeServerApp, the client checks its local cache. Similarly, before the OrgTreeServerApp passes client requests to the OrgTreeProvider, it checks its local cache. If the cache does not contain the data to fulfill the request, the request is passed to the OrgTreeProvider, and the results are then cached.

The client cache period is set by the following eclipse preference:

com.ibm.collaboration.realtime.directory.sametime/orgTreeCacheExpirationTime=<millis>

The client cache can also be reset at anytime by the user in the Organization View UI.

The server side cache period can be set in the service configuration file, st.orgtree.properties,

using the “orgTreeCacheExpirationTime” property.

The default for both client and server is 604800000 milliseconds, which equals seven days.

Chapter 3. Organization tree providers

A key interface in the organization tree service is the OrgTreeProvider interface, which defines the APIs used to create the organization view from the directory. The organization tree service includes two OrgTreeProvider implementations, a directory information tree (DIT) based provider and an attribute based provider.

The DIT based provider should be used if the organization tree should mirror the directory information tree itself. With this provider, the Sametime client UI will show the directory tree much like a typical LDAP client would when browsing the DIT.

The attribute based service should be used if the organization tree should reflect hierarchical relationships between directory entries as indicated by certain attributes.
In order to use the attribute based model, the following two types of directory attributes are necessary:

1. An attribute which defines the distinguished name (DN) of the entry's parent. For example a "manager" attribute to indicate the manager of that entry.

2. An attribute which defines whether or not an entry is a parent to other entries. For example, an "isManager" attribute to indicate whether or not an entry is a manager.

If neither the DIT or attribute based OrgTreeProvider will work for the directory in question, a custom OrgTreeProvider implementation can be created.

Chapter 4. Deploying the organization tree service

1. Fill in the organization tree configuration properties, st.orgtree.properties according to your directory requirements. The st.orgtree.properties file can be found in the sametime.orgtree.service kit. The file includes key properties and a description of each. Additionally, properties are outlined below in the “Key organization tree service properties” section.

2. Validate the organization tree service. This must be done prior to deploying the service. The "orgTreeValidate" script in the SDK will begin the validation routine. The OrgTreeValidator will confirm that the organization tree service is properly configured through st.orgtree.properties such that it can generate the complete organization tree. For details see the "Validating the organization tree service" section below.

3. Modify sametime.ini on the Community Server to allow connections from the organization

tree SA. Once validation is complete, the organization tree service can be deployed. See "How to enable the organization tree service" below.

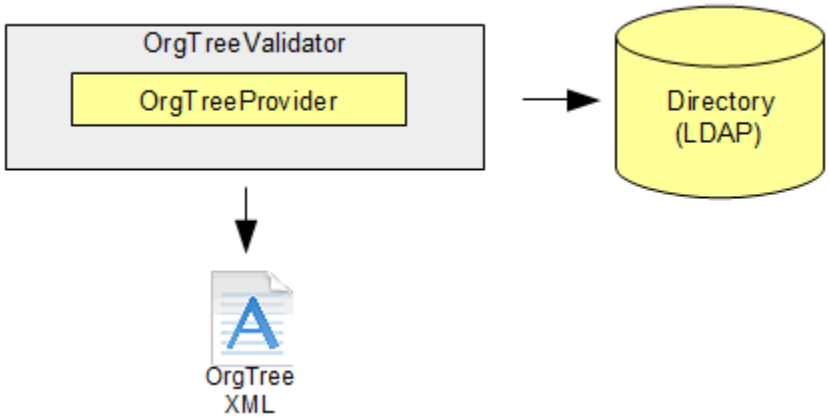
4. Start the organization tree service application, OrgTreeServerApp. The OrgTreeServerApp is started by running the "orgTreeServerApp" script.

Chapter 5. Validating the organization tree service

To ensure the successful operation of the organization tree service, it is crucial to validate the organization tree service prior to deployment. The validation process will run the organization tree service against the directory according to the configuration properties from st.orgtree.properties and will generate a report of results and statistics, as well as an XML file representation of the organization tree. The organization tree service should not be deployed without first completing organization tree validation. To run the organization tree validator follow these steps:

1. Run the "orgTreeValidate" script. The validator will run and traverse the directory from the “rootDn” property specified in st.orgtree.properties as it creates the organization tree. This can take some time depending on the number of items in the directory. For example, to process 150,000 records might take around 3.5 hours, depending on bandwidth and directory performance. Before doing a full validation run, it is possible to do a small run to get a quick sense of how the OrgTreeProvider is doing with the directory. You can set the “maxOrgTreeValidationErrors” or “maxOrgTreeValidationNodes” properties in order to halt validation after a certain number of errors occur or a certain number of nodes are processed (respectively). For more, see the table of organization tree service properties below.

2. Review the organization tree validation report. The validator will create an "orgtree-validator" directory containing a log file, a text report of statistics and errors, and a complete organization tree XML file showing the structure of the organization tree. Review the text report for errors and open the XML file in a browser to confirm the tree looks as expected. If there are no errors and the organization tree looks correct, validation is complete.



Chapter 6. How to enable the organization tree service

In order for the organization tree service to log in to the Sametime server, the IP address of the machine running the service must be in the server's trusted IP list. This can be done by using one of the following options:

- Add your IP to the server trusted IP list. To do so, open stconfig.nsf and add the IP to the CommunityTrustedIPS field in the CommunityConnectivity document. Separate multiple IP addresses with commas or semicolons.

- Add the following line to the Debug section of the Sametime.ini file:
  VPS_BYPASS_TRUSTED_IPS=1
  This approach is not secure and should only be used for test purposes outside of the production environment.

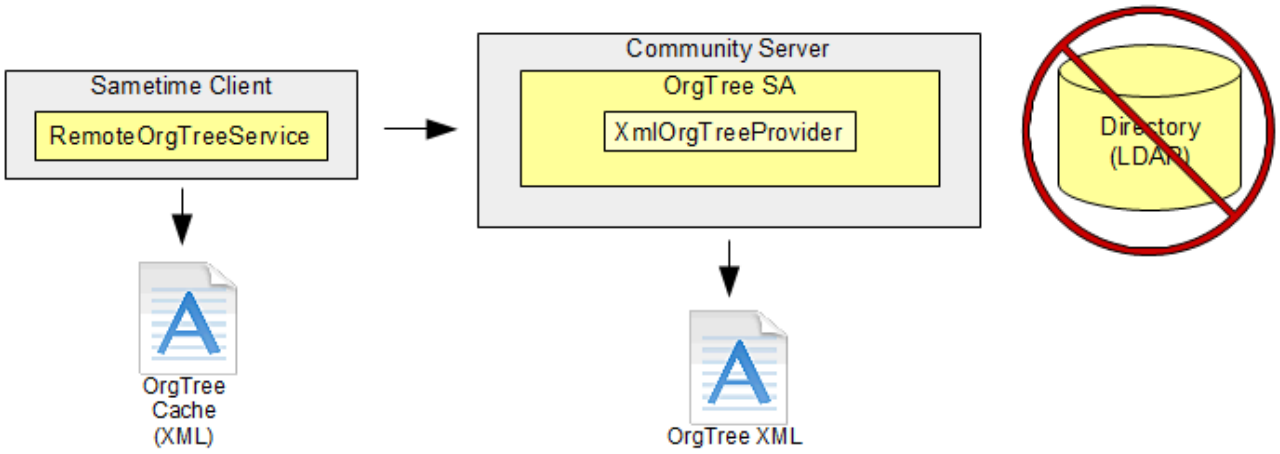
Chapter 7. Deployment alternatives

XML organization tree provider

The XML file created during the validation of the organization tree can be used as a datasource, thereby preventing any load on the directory from client requests. The organization tree service includes an XMLOrgTreeProvider implementation for this purpose. In order to enable this OrgTreeProvider, the following properties must be set in st.orgtree.properties:

orgTreeProvider=com.ibm.collaboration.realtime.orgtree.XmlOrgTreeProvider
xmlOrgTreePath=<file-path-to-my-orgtree-xml>

In oder to keep this type of deployment up to date, the OrgTreeValidator can be scheduled to run during off-peak hours in order to generate updated versions of the XML (for example using a system scheduler task). The OrgTreeServerApp can then be restarted in order to load the new XML file.



Chapter 8. Key organization tree service properties

Table listing key st.orgtree.properties

Property	Description
host	The host name of the community server to which the OrgTreeServerApp connects.
orgTreeCacheExpirationTime	The orgtree cache expiration time in milliseconds. The orgTreeCacheExpirationTime is used to determine if a cached organizational node is expired or not. If a node is expired, a remote call is made to fulfill the query and the results are then added to the cache.
orgTreeCacheDirPath	The path to the directory where the orgtree cache file is located. Not required, by default the directory from which the application is run is used.
orgTreeCacheFileName	The name of the orgtree cache file. By default the name is orgtree-cache.xml
java.naming.provider.url	The LDAP URL of the directory server
java.naming.security.authentication	JNDI security property. See http://download.oracle.com/javase/1.5.0/docs/guide/jndi/spec/jndi/properties.html
orgTreeProvider	The fully qualified name of the class that provides the organization tree. Use either the out-of-the-box DitOrgTreeProvider, AttributeBasedOrgTreeProvider, XmlOrgTreeProvider, or a custom provider. The DitOrgTreeProvider should be used if the structure of the organization tree comes from the DIT (directory information tree) itself. The AttributeBasedOrgTreeProvider should be used if the structure of the organization tree is determined by attributes that define hierarchical relationships between directory items. The XmlOrgTreeProvider should be used if using a static orgtree XML file created by the orgtree validation process.
orgNodeProvider	Optional property specifying the fully qualified name of a custom OrgNodeProvider implementation. If not specified, a default OrgNodeProvider is used.
rootDn	The root distinguished name to use in the client organization tree. Can be a single rootDn or a semi-colon delimited list of multiple root DNs. For example, rootDn=dc=foobar,dc=com
includeRootDn	Whether or not to include the root DN in the organization tree. If false, the subnodes of the root DN appear as the root nodes
ldapSearchBase	The search base to use when searching the directory. For example "o=acme.com". This property should be specified for the attribute based provider. For the DIT provider, if not specified, the rootDn is used.
groupObjectClasses	The object classes in the directory that specify a group type organizational node. For example: “organizationalUnit,organization,domain”.
personObjectClasses	The object classes in the directory that specify a person type organizational node. For example: “person”.
contactIdAttrib	The attribute that represents the Sametime contact ID of a person entry. If not specified, the DN of the entry is assumed and used. If this attribute is incorrect, there will be no awareness in the organization view and the locate user in organization view feature will not work.

personDisplayNameAttrib	The display name attribute to use for a person organizational node.
groupDisplayNameAttrib	The display name attribute to use for a group organizational node.
ldapSearchReturningAttribs	An optional comma delimited list of extra directory attributes to return. Only needed if using a custom OrgNodeProvider. For example, if extending DefaultOrgNodeProvider to append a custom attribute to the display name, define that attribute here to ensure that it is included in the list of attributes returned from the directory. The default returned attributes includes the object class, the contact ID attribute (if specified by the contactIdAttrib property), the person display name attribute (as specified by the personDisplayNameAttrib property), and the group display name attribute (as specified by the groupDisplayNameAttrib property).
parentDnAttrib	In an attribute based organization tree, the directory attribute of an entry that specifies that entry's parent DN. For example, a "manager" attribute. Required for AttributeBasedOrgTreeProvider, not required if using the DitOrgTreeProvider.
isParentAttrib	In an attribute based organization tree, the directory attribute of an entry that indicates if that entry is a parent of other person entries. For example, an "ismanager" attribute. Required for AttributeBasedOrgTreeProvider, not required if using the DitOrgTreeProvider.
excludedDns	A semi-colon delimited list of distinguished names to exclude from the organization tree. Note that the children of excluded entries are also excluded.
maxOrgTreeValidationErrors	The number of validation errors that can occur before the organization tree validator stops. Default is -1, meaning no limit.
maxOrgTreeValidationNodes	The max number of organization nodes to validate before stopping validation. The maxOrgTreeValidationNodes prop can be set to stop validation early because validating a large tree, for example 100,000 nodes or more, can take a few hours. So for example, when validating a very large directory the first time, maxOrgTreeWalkerNodes can be set to a low value such as 1000 in order to produce a quick initial validation assessment.
orgTreeWalkerGetChildrenDelay	The delay in milliseconds to pause between calls to retrieve the children of a directory entry when walking the directory in order to create the organization tree as part of the validation process. This can be used lighten the impact of validation on the directory server at the expense of speed.

Chapter 9. Key classes and interfaces

Table listing key classes and interfaces of the organization tree service

Class	Description
OrgTreeServerApp	The main entry point of the organization tree service application. This class connects to the community server and invokes the OrgTreeProvider APIs to return the organization tree to the Sametime client. This is not an extendable class.
OrgTreeProvider	The OrgTreeProvider interface is the key interface of the organization tree service. It is responsible for searching the directory in order to build the tree and locate contacts within the tree. A custom OrgTreeProvider can be implemented if DIT or attribute based OrgTreeProvider is not suited to the directory to be used.
OrgNodeProvider	The OrgNodeProvider is used by the OrgTreeProvider to create OrgNode objects from directory search results and can be thought of as a factory for OrgNode objects.

	Custom OrgNodeProviders can be implemented if the DefaultOrgNodeProvider cannot be extended for some reason.
DefaultOrgNodeProvider	The default OrgNodeProvider implementation. This class can be extended to customize aspects of the OrgNode objects returned to the OrgTreeProvider. For example, the display name, attributes, and sort order of the OrgNode objects can be customized.
OrgNode	The OrgNode is an abstraction for an entry in the directory. An OrgNode can represent either a person or a group.
OrgTreeWalker	The OrgTreeWalker interface is used by the organization tree validation process. An OrgTreeProvider must also implement OrgTreeWalker so that it can be validated by the OrgTreeValidator. The OrgTreeWalker is passed an OrgTreeWalkerHandler, which provides a callback mechanism for the OrgTreeValidator to keep track of the tree. The OrgTreeWalker must call the methods of the OrgTreeWalkerHandler as it walks the organization tree.
OrgTreeWalkerHandler	An interface implemented by the OrgTreeValidator, so that it can receive callbacks as the OrgTreeProvider traverses the organization tree.

In order to customize UI aspects of the organization tree nodes, such as dispay name, sort order, or filtering of results, the DefaultOrgNodeProvider can be subclassed and the appropriate methods overridden. The OrgNodeProvider class to use is specified by the “orgTreeProvider” property in st.orgtree.properties. In order to customize how results are retrieved from the directory in order to build the tree structure, the OrgNodeProvider interface can be implemented. The OrgNodeProvider to use is specified by the “orgNodeProvider” property in st.orgtree.properties. For more information on these classes, see the organization tree javadoc.

Chapter 10. Developing a custom provider

Using the Eclipse IDE, import the organization tree service project found in the SDK's server\commserver\orgtree\service directory. Use the Import “Existing Projects into Workspace” option and select the “service” directory. The project “orgtree.svc” should appear in your workspace. Next import the organization tree sample project found in the SDK 's server\commserver\orgtree\sample directory. Again, use the Import “Existing Projects into Workspace” option. The project “orgtree.sample” should appear in the workspace. The orgtree.sample project depends on the libraries exported by the orgtree.svc sample. Using this sample, you can better understand how to extend the DefaultOrgNodeProvider in order to override the OrgNodeProvider's “getDisplayName” and “exclude” APIs.

In order to deploy the customized sample, right click the eclipse project and use the Export menu to export a jar file into the orgtree service directory. Next modify the orgTreeServerApp script so that the jar file is part of the classpath. For example, if you exported orgtree.sample.jar, in orgTreeServerApp.bat list the jar like so:

```
set CP="orgtree.svc.impl.jar;orgtree.svc.api.jar;%STJAVATK_JAR%;orgtree.sample.jar"
```

Next modify st.orgtree.properties to reference the custom OrgNodeProvider. For example:

```
orgNodeProvider=com.ibm.rtc.orgtree.sample.SampleOrgNodeProvider
```

After starting the organization tree service, the custom OrgNodeProvider will be used.

Chapter 11. FAQ

How can I customize the display names of the OrgNodes shown in the client UI? For example, I would like to include a telephone number after the name.
This can be done by filling in the property “ldapSearchReturningAttribs” in st.orgtree.properties with the appropriate phone number directory attribute and by extending the DefaultOrgNodeProvider's getDisplayName() method in order to access and append the telephone number attribute to the result.

OK, but I would like to customize the name with an attribute from a third party datasource, not LDAP. Can I do that?
A custom OrgNodeProvider can access any third party datasource that it needs.

Can I exclude certain people from the results?
The st.orgtree.properties contain an “excludedDns” property that can be used to specify a static list of distinguished names to exclude from directory search results. Additionally, the OrgNodeProvider contains an “exclude” API that is called by the OrgTreeProvider when processing results from the directory or cache This method includes the requestor's Sametime contact ID which can serve as the basis for the exclusion. The DefaultOrgNodeProvider can be extended in order to override the exclude method. Note that when an entry is excluded, any children of that entry are also excluded. Another way to go about excluding certain nodes, especially if the entries are near the root of the tree, is to specify multiple rootDNs in st.orgtree.properties where the rootDNs are beneath the entries to exclude.

How can I sort the items in the organization view?
The OrgNodeProvider contains a getOrgNodeComparator() method that returns a java.util.Comparator used to sort OrgNode objects. The DefaultOrgNodeProvider can be extended in order to override this method. The Comparator can sort the OrgNode results by type (group, person) as needed. By default groups are sorted ahead of persons, and within each type, sorted alphabetically. If OrgNode objects are to be sorted by a custom OrgNodeProvider using additional directory attributes, the "ldapSearchReturningAttribs" property needs to be set in st.orgtree.properties. This will ensure the OrgNode objects contain these directory attributes for the custom sorter. Note that all sorting is done at the organization tree service level; there is no locale based sorting at the client level.

How can I provide different organization views to different users?
The OrgTreeService is a remote service that connects to LDAP as a single user on behalf of all client requests. Therefor the view of the directory is from the perspective of the principal that is used to connect to the directory. However, it is possible to filter results for different users using a custom OrgNodeProvider. The DefaultOrgNodeProvider can be extended and the exclude API overridden to filter results based on the person making the request.

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.
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
5 Technology Park Drive

Westford Technology Park
Westford, MA 01886

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 information 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. All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

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

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.

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. _enter the year or years_. All rights reserved.

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

Trademarks

These terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM

AIX

DB2

DB2 Universal Database Domino

Domino

Domino Designer

Domino Directory

i5/OS

iSeries

Notes

OS/400

Sametime

System i

WebSphere

AOL is a registered trademark of AOL LLC in the United States, other countries, or both.

AOL Instant Messenger is a trademark of AOL LLC in the United States, other countries, or both.

Google Talk is a trademark of Google, Inc, in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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