Netcool/OMNIbus Probe for CA Spectrum (CORBA)

Version 1.3

Reference Guide

August 28, 2009

SC23-8593-01

Netcool/OMNIbus Probe for CA Spectrum (CORBA)

Version 1.3

Reference Guide

August 28, 2009

SC23-8593-01

Note
Before using this information and the product it supports, read the information in "Notices and Trademarks," on page 17.

Edition notice

This edition applies to version 0.0.6458 of Probe for CA Spectrum (CORBA) (SC23-8593-01) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC23-8593-00.

Contents

Document control page

Use this information to track changes between versions of this guide.

The Probe for CA Spectrum (CORBA) documentation is provided in softcopy format only. To obtain the most recent version, visit the IBM® Tivoli® Knowledge Center:

http://www-01.ibm.com/support/knowledgecenter/?lang=en#!/SSSHTQ/omnibus/probes/common/Probes.html

Table 1. Document modification history

Document version	Publication date	Comments
SC23-8593-00	December 19, 2008	First IBM publication.
SC23-8593-01	August 28, 2009	Support for Linux added.

IBM Tivoli Netcool/OMNIbus Probe for CA Spectrum (CORBA)

The SpectroSERVER acts as a network management server for the Spectrum Enterprise Manager. The Spectrum Enterprise Manager is an element manager that uses the SpectroSERVER to manage CA Aprisma and third party devices.

The Probe for CA Spectrum (CORBA) collects alarm information from the SpectroSERVER and sends them to Netcool/OMNIbus.

This guide contains the following sections:

Summary

Each probe works in a different way to acquire event data from its source, and therefore has specific features, default values, and changeable properties. Use this summary information to learn about this probe.

The following table provides a summary of the Probe for CA Spectrum (CORBA).

Table 2. Summary

Probe target	CA Spectrum version 8.x **Note:** This probe does not support CA Spectrum version 9.
Probe executable name	`nco_p_spectrum_corba`
Patch number	1.3
Properties file	`$OMNIHOME/probes/`*`arch`*`/spectrum_corba.props`
Rules file	`$OMNIHOME/probes/`*`arch`*`/spectrum_corba.rules`
Connection method	CORBA
Remote connectivity	The Probe for CA Spectrum (CORBA) can connect to a remote device using a CORBA interface.
Internationalization	Available **Note:** The probe supports internationalization on IBM Tivoli Netcool/OMNIbus V7.3.0, 7.3.1 or 7.4.0.
Peer-to-peer failover functionality	Available
IP environment	IPv4 and IPv6 **Note:** The probe is supported on IPv6 when running on IBM Tivoli Netcool/OMNIbus V7.3.0, 7.3.1 and 7.4.0 on all UNIX and Linux operating systems.

Table 2. Summary (continued)

Federal Information Protocol Standards (FIPS)	IBM Tivoli Netcool/OMNIbus uses the FIPS 140-2 approved cryptographic provider: IBM Crypto for C (ICC) certificate 384 for cryptography. This certificate is listed on the NIST website at http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2004.htm. For details about configuring Netcool/OMNIbus for FIPS 140-2 mode, see the *IBM Tivoli Netcool/OMNIbus Installation and Deployment Guide*.

Configuring the probe

After installing the probe, you need to make various configuration settings to suit your environment.

The following topics describe the configuration that you need to perform:

- "Configuring the SpectroSERVER"
- "Receiving probable cause details"
- "Setting probe environment variables" on page 3

Configuring the SpectroSERVER

SpectroSERVER should accept the CORBA connections from the system and the userid that run the probe. The configuration includes the following:

1. Creating a Spectrum User Model with the userid that runs the probe
2. Using SpectroGRAPH or OneClick to add details of the system that runs the probe in the server list of the SpectroSERVER
3. Adding this userid to the associated UserList

For elaborate details on configuring the SpectroSERVER, refer to the Spectrum's *CORBA API Programmer Guide*, *Distributed SpectroServer Guide*, and the *Installation Guide*. The *nsutil* utility described in the *CORBA API Programmer Guide* helps you confirm whether the CORBA services are available.

Receiving probable cause details

The probable cause script enables the event enrichment feature of the probe and adds the probable cause information with alarms. The script when run on the SpectroSERVER generates a lookup file; this lookup file matches a spectrum cause code with a meaningful description of the causes that triggered the event. The probe retrieves the probable cause information for various types of alarms from this lookup file.

To generate the lookup file do the following:

1. Run the probable cause script on the SpectroSERVER using the following command:

   perl SpectrumCause.pl <*Spectrum install directory*> <*temporary location for generated lookup file*>

   This command generates the lookup file named SpectrumCause.lookup in the temporary directory specified.
2. Copy the lookup file to the $OMNIHOME/probes/*arch* directory of the system that runs the probe.

3. To make the rules file access the lookup file in this location, uncomment the following two lines of the spectrum_corba.rules file:

```
include "spectrum-Cause_Details.lookup"
 [@EventId, $Temp_Summary] = lookup($CauseNum, spectrum-Cause_Details)
```

4. Restart the probe.

Note: The path to this file is included in the rules file. If you are not running the probe from $OMNIHOME/probes/*arch*/, you must change this path accordingly.

Setting probe environment variables

To run the probe from a different environment, you should set its environment variables and enable the probe access the required jar files.

The probe automatically adds the following jar files in the $OMNIHOME/probes/java directory in its CLASSPATH:

- `global81.jar`
- `ssorb81.jar`
- `utilnet81.jar`
- `utilsrv81.jar`
- `vbhelper.jar`
- `ssorbutil81.jar`

Add the following jar file from the SPECTRUM install at *$SPECROOT*/lib to the CLASSPATH:

- `util81.jar`

Tip: Copying the util81.jar to the $OMNIHOME/probes/java directory makes the probe automatically add the jar file.

You must add the path to the Java 5 bin directory in the $PATH environment variable.

Internationalization support

The probe supports multibyte character sets (for example, Japanese) and character sets that contain individual multibyte characters (for example German, French, and Spanish). To view the character sets correctly, you must configure the locale settings on the host machine correctly.

If you are using a language that contains multibyte characters, you must set the LANG environment variables to the name of your character set, and export the LC_ALL environment variable. For example, if you are using Japanese, set these environment variables to ja_JP.UTF-8; if you are using German, set these environment variables to de_DE.UTF-8. This will enable the probe to recognise the multibyte characters used by your character set when they occur in any network events.

The probe supports the following language locales:

Table 3. Supported language locales

Languages	AIX	HP-UX	Solaris	Linux
English (US)	en_US	en_US	en_US	en_US

Table 3. Supported language locales (continued)

Languages	AIX	HP-UX	Solaris	Linux
Simplified Chinese	zh_CN	zh_CN	zh_CN	zh_CN
Traditional Chinese	zh_TW	zh_TW.eucTW	Zh_TW.big5	zh_TW.big5
Czech	cs_CZ	cs_CZ	cs	cs_CZ
French (standard)	fr_FR	fr_FR	fr	fr_FR
German (standard)	de_DE	de_DE	de	de_DE
Hungarian	hu_HU	hu_HU	hu	hu_HU
Italian (standard)	it_IT	it_IT	it	it_IT
Japanese	ja_JP	ja_JP	ja	ja_JP
Korean	ko_KR	ko_KR	ko	ko_KR
Polish	pl_PL	pl_PL	pl	pl_PL
Portuguese (Brazilian)	pt_BR	pt_BR	pt	pt_BR
Russian	ru_RU	ru_RU	ru	ru_RU
Spanish	es_ES	es_ES	es	es_ES

Example multi-byte character set on Solaris

The following steps describe how to configure Solaris to use the Japanese character set:

1. Install the necessary components for Japanese on to the host machine using the Solaris CD.
2. Set the `LANG` and `LC_ALL` environment variables to `ja_JP PCK`. This uses SJIS encoding.

   Note: You may have to set the `LANG` in the host machine's default settings file and reboot it to make the changes take effect.
3. Make sure that the file `$OMNIHOME/platform/arch/locales/locales.dat` has the following entry:

   ```
   locale = ja_JP PCK, japanese, sjis
   ```

   Where `ja_JP PCK` is the vendor locale, `japanese` is the Sybase language, and `sjis` is the Sybase character set.

Peer-to-peer failover functionality

The probe supports failover configurations where two probes run simultaneously. One probe acts as the `master` probe, sending events to the ObjectServer; the other acts as the `slave` probe on standby. If the master probe fails, the slave probe activates.

While the slave probe receives heartbeats from the master probe, it does not forward events to the ObjectServer. If the master probe shuts down, the slave probe stops receiving heartbeats from the master and any events it receives thereafter are forwarded to the ObjectServer on behalf of the master probe. When the master probe is running again, the slave probe continues to receive events, but no longer sends them to the ObjectServer.

Example property file settings for peer-to-peer failover

You set the peer-to-peer failover mode in the properties files of the master and slave probes. The settings differ for a master probe and slave probe.

Note: In the examples, make sure to use the full path for the property value. In other words replace `$OMNIHOME` with the full path. For example: `/opt/IBM/tivoli/netcool`.

The following example shows the peer-to-peer settings from the properties file of a master probe:

```
Server      : "NCOMS"
RulesFile   : "master_rules_file"
MessageLog  : "master_log_file"
PeerHost    : "slave_hostname"
PeerPort    : 6789 # [communication port between master and slave probe]
Mode        : "master"
PidFile     : "master_pid_file"
```

The following example shows the peer-to-peer settings from the properties file of the corresponding slave probe:

```
Server      : "NCOMS"
RulesFile   : "slave_rules_file"
MessageLog  : "slave_log_file"
PeerHost    : "master_hostname"
PeerPort    : 6789 # [communication port between master and slave probe]
Mode        : "slave"
PidFile     : "slave_pid_file"
```

Data acquisition

Each probe uses a different method to acquire data. Which method the probe uses depends on the target system from which it receives data.

The Probe for CA Spectrum (CORBA) gathers events from the CA SpectoSERVER using a Common Object Request Broker Architecture (CORBA) interface. CORBA is an Object Management Group specification that provides a standard interface definition between objects in a distributed environment; that is, it allows applications to communicate with one another regardless of where they are located or who has designed them.

Data acquisition is described in the following topics:

- "Connecting to the CORBA interace"
- "Re-synchronisation"

Connecting to the CORBA interace

The Probe for CA Spectrum (CORBA) connects to the SpectoSERVER using a CORBA interface, and listens for spectrum alarm updates. The probe also polls the Spectrum CORBA interface to ensure that the connection with the SpectroSERVER is active.

When the probe starts, it connects to the SpectroSERVER using the **SpectroServerName** and **SpectroServerDomain** properties. If the probe and SpectroSERVER run in different systems from different broadcast domains, then also specify the system that runs the SpectroSERVER using the **SpectroServerIP** property. On connection to the server, the probe listens for new alarms from the SpectroSERVER.

The CORBA status monitor that the probe initializes monitors the status of the CORBA services. The probe checks the status of the CORBA services at intervals specified by the **PollingInterval** property. If the SpectroSERVER disconnects from the CORBA interface, the status monitor reports an error and shuts down the probe.

If SpectroSERVER is configured with failover feature, the probe can connect to the secondary SpectroSERVER in failover mode and continue to process alarms. When processing alarms the probe logs whether it is connected to the primary SpectroSERVER or the secondary SpectroSERVER.

Re-synchronisation

The probe initially receives a list of all active alarms from the SpectroSERVER.

On reconnection, if the **AllAlarmsOnRestart** property is set to 1, the probe listens for any alarm updates, and receives all the active alarms from the SpectroSERVER. A value of 1 also sends the already received events. If the **AllAlarmsOnRestart** property is set to 0, the probe only retrieves the alarms generated while the probe was disconnected.

Ensure that the rules file is configured to maintain the `Tally` field of the ObjectServer.

Note: Events deleted from the SpectroSERVER while the probe is disconnected remain in the ObjectServer, and must be cleared out manually. If the probe uses a

timestamp file, some event duplication of alarms may occur.

Updating SpectroSERVER fields

The Probe for CA Spectrum (CORBA) is supplied with a Command Line Interface (CLI). This interface allows you to execute commands using the probe.

Using the CLI the probe can update the SpectroSERVER fields.

The following table shows the fields that the probe can update:

Table 4. SpectroSERVER fields

Field	Description
Acknowledged	This field indicates whether the alarm has been acknowledged by a Spectrum operator.
Alarm Status	This field indicates status of the corrective action.
Event ID List	This field lists all the Spectrum events that resulted in the alarm.
Troubleshooter	This field shows the name of the troubleshooter.
TroubleshooterModel	This field identifies the type of the troubleshooter.
TroubleTicketID	This field contains the associated trouble ticket number for the given alarm.

Note: Since the CLI is based upon TCP/IP connections, you can connect to the probe from anywhere.

Command line interface

To use the CLI, you must use the **CommandPort** property to specify a port to which commands will be sent. When you want to execute commands, telnet to this port. The **CommandPortLimit** property determines the maximum number of telnet connections that can be made to the probe.

Note: You can use Netcool/OMNIbus automations to update the ObjectServer fields with corresponding updates in the SpectroSERVER fields. For details, see the *IBM Tivoli Netcool/OMNIbus Administration Guide*, (SC23-6371).

The following table describes the commands that you can use with the CLI.

Table 5. CLI commands

Command	Description
acknowledgeAlarm *alarmID*	Use this command to acknowledge an alarm in the SpectroSERVER, where *alarmID* is the identifier of the alarm being acknowledged.
unacknowledgeAlarm *alarmID*	Use this command to unacknowledge an alarm in the SpectroSERVER, where *alarmID* is the identifier of the alarm being unacknowledged.
clearAlarm *alarmID*	Use this command to clear the given alarm identifier in the SpectroSERVER, where *alarmID* is the identifier of the alarm being cleared.
updateStatus *alarmID status*	Use this command to update the alarm with the status specified.

Table 5. CLI commands (continued)

Command	Description
updateTroubleShooterName *alarmID name*	Use this command to update the alarm with the provided trouble shooter name.
updateTroubleTicket *alarmID ticketID*	Use this command to update the alarm with the trouble ticket identifier specified.
updateTroubleShooterModel *alarmID model*	Use this command to update the alarm with the model type of the troubleshooter specified.
updateEventList *alarmID eventIDList*	Use this command to update the alarm with the event list specified.

Properties and command line options

You use properties to specify how the probe interacts with the device. You can override the default values by using the properties file or the command line options.

The following table describes the properties and command line options specific to this probe. For information about default properties and command line options, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*, (SC23-6373).

Table 6. Properties and command line options

Property name	Command line option	Description
AllAlarmsOnReStart *integer*	`-allalarmsonrestart` *integer*	Use this property to specify whether the probe retrieves all active alarms, or only those that have been created since the last alarm that the probe received. `0`: The probe only retrieves alarms generated since disconnection. `1`: The probe collects all active alarms from SpectroSERVER. The default is `0`.
CommandPort *integer*	`-commandport` *integer*	Use this property to specify the port to which users can telnet to communicate with the CORBA interface using the Command Line Interface (CLI) supplied with the probe. For details about the CLI, see "Updating SpectroSERVER fields" on page 7. The default is `7777`.
CommandPortLimit *string*	`-commandportlimit` *integer*	Use this property to specify the maximum number of telnet connections that the probe can make. The default is `10`.

Table 6. Properties and command line options (continued)

Property name	Command line option	Description
SpectroServerDomain *string*	-spectroserverdomain *string*	Use this property to specify the name of the system that runs the SpectroSERVER. The default is " ". **Note:** This is the non-fully qualified hostname of the system that runs the SpectroSERVER; and it is always specified in lowercase.
SpectroServerIP *integer*	-spectroserverip *integer*	Use this property to specify the IP of the system that runs the SpectroSERVER. The default is 0.0.0.0. **Note:** This property is required only when the systems that run the probe and SpectroSERVER are in different broadcast domains.
SpectroServerName *string*	-spectroservername *string*	Use this property to specify the hostname of the system where the SpectroSERVER is installed. The default is " ".
SpectroServerPoll Interval *integer*	-spectroserverpoll interval *integer*	Use this property to specify the time (in seconds) the probe waits before polling the CORBA interface between successive polls of the SpectroSERVER. The default is 20.
TimeStampFile *integer*	-timestampfile *integer*	Use this property to specify a file in which the probe stores the timestamp of the last alarm. The default is " ".

Elements

The probe breaks event data down into tokens and parses them into elements. Elements are used to assign values to ObjectServer fields; the field values contain the event details in a form that the ObjectServer understands.

The following table describes the elements that the Probe for CA Spectrum (CORBA) generates. Not all the elements described are generated for each event; the elements that the probe generates depends upon the event type.

Table 7. Elements

Element name	Element description
$Acknowledged	This element indicates if the alarm has been acknowledged by a Spectrum operator.

Table 7. Elements (continued)

Element name	Element description
$AlarmID	This element contains the identifier of the alarm. **Note:** The alarm ID changes when events failover to a secondary SpectroSERVER.
$ClearMe	This element indicates whether the event is a resolution event. A value of true sets the Object Server field @Type to 2 in the rules file; this indicates that it is a resolution event.
$CauseNum	This element shows the probable cause number of the event.
$CreationDate	This element contains the date when the alarm was generated.
$EventIDList	This element contains the list of Spectrum event IDs that triggered the alarm.
$ModelID	This element shows the identifier of the model.
$ModelName	This element contains a description of the model.
$ModelType	This element shows the type of the device that raised the alarm.
$PrimaryAlarm	This element indicates the priority of this alarm in SpectroSERVER. A value of true identifies the alarm as a primary alarm. Primary alarms have the highest priority.
$Pre-existing	This element indicates whether the received event is already present in the SpectroServer.
$Primary	This element indicates whether the probe is connected to the primary SpectroSERVER or to a secondary SpectroSERVER. **Note:** A value of false indicates that the probe is connected to a secondary server.
$Priority	This element indicates the priority in the SpectroSERVER.
$Severity	This element indicates the severity of the alarm in the SpectroSERVER.
$Source	This element indicates whether the alarm is generated before failover.
$Status	This element displays the status of the event.
$Tally	This element shows the number of times the event has occurred.

Error messages

Error messages provide information about problems that occur while running the probe. You can use the information that they contain to resolve such problems.

The following table describes the error messages specific to this probe. For information about generic error messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide,* (SC23-6373).

Table 8. Error messages

Error	Description	Action
Initial CORBA connection errors:		

`Exception when attempting to run status monitor Probe shutting down because CORBA Status monitor not polling successfully.`
 	
`Exception thrown when attempting to start CORBA service Probe shutting down as unable to access CORBA service, please ensure that server configuration is complete and try again Probe shutting down because Probe unable to access CORBA service`
 	
`Exception when attempting to establish a filter for alarms Probe shutting down as unable to receive alarms,please ensure that the timestamp file pointed to does not contain corrupted data Probe shutting down because unable to establish Alarm Watch filter`
 	
`Exception when attempting to establish an Alarm Watch Probe shutting down as unable to receive alarms,please ensure that server configuration is complete and try again Probe shutting down because unable to establish Alarm Watch`
	The probe could not initialize connection with the CORBA interface and is shutting down.	Check the SpectroSERVER configuration.
		
		Check the values set for the **`SpectroServerDomain`**, **`SpectroServerName`**, and **`SpectroServerIP`** properties.
		
		Check the firewall settings in your environment.
		
		Check that the appropriate additions added to the server list of the SpectroSERVER.
		
		Check that the UserModel with the userid that runs the probe has been created on the SpectroSERVER.

Failure of CORBA connection:		

Table 8. Error messages (continued)

Error	Description	Action

`Error occurred attempting to pass on details of cleared alarms to the Object Server.`
`Probe shutting down because Problem processing cleared alarms.`
 	
`Error occurred attempting to pass updates to the Object Server`
`Probe shutting down because Problem processing updated alarms.`
 	
`Error occurred attempting to pass details of new alarms to the Object Server`
`Probe shutting down because Problem processing new alarms`
 	
`Error in connecting to CORBA interface`
`Error external to CORBA framework`
`Probe shutting down because External error in connection to CORBA interface`
 	
`Exception thrown when querying whether we are accessing Primary SpectroSERVER`
`Probe shutting down because Problem querying SpectroSERVER.`
 	
`Connection to CORBA interface has been lost.`
`Error internal to CORBA interface`
`Probe shutting down because Internal error in CORBA interface, connection lost`
 	
`Error shutting down connection to CORBA interface`
	The probe connection with the CORBA interface has failed.	Restart the probe.
		
		Check whether the SpectroSERVER configurations are changed.
		
		Check whether the SpectroSERVER is running.

Timestamp file errors:		

Table 8. Error messages (continued)

Error	Description	Action

`Failed to pick up probe property TimeStampFile Not using timestamp file, proceeding to collect all active alarms.`
 	
`Timestamp of alarms not stored for re-sync`
 	
`Unable to create timestamp file`
 	
`Unable to find time stamp file`
 	
`Unable to write to timestamp file`
 	
`Error closing timestamp file`
 	
`Error opening timestamp file for reading`
	The probe failed to use the timestamp file. These errors are not fatal error messages.	Check the specific message and confirm that the timestamp file is accessible

Command port errors: (They occur only when the command port functionality is used)		

Table 8. Error messages (continued)

Error	Description	Action

`Alarm + alarmID + Error clearing alarm`
`Error clearing alarm`
 	
`Alarm + alarmID + Error in Event ID List update`
`Error updating the event ID List`
 	
`Alarm + alarmID + Error in Trouble Shooter Model update`
`Error updating the trouble shooter model`
 	
`Alarm + alarmID + Error in Trouble Ticket update`
`Error updating the trouble ticket`
 	
`Alarm + alarmID + Error in Troubleshooter Name update`
`Error updating the trouble shooter name`
 	
`Alarm + alarmID + Error in status update`
`Error updating the alarm status`
 	
`Alarm + alarmID +Error in unacknowledge`
`Error un-acknowledging alarm`
 	
`Alarm + alarmID + Error in acknowledge`
`Error acknowledging alarm`
	The probe failed to update the event.	Check that the alarm you are trying to update exists in the SpectroSERVER.
		
		Note: If the alarm is not found, the SpectroSERVER shows an error message.

ProbeWatch messages

During normal operations, the probe generates ProbeWatch messages and sends them to the ObjectServer. These messages tell the ObjectServer how the probe is running.

The following table describes the raw ProbeWatch error messages that the probe generates. For information about generic ProbeWatch messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*, (SC23-6373).

Table 9. ProbeWatch messages

ProbeWatch message	Description	Triggers/causes
`Probe has lost connection to CORBA interface, error external to CORBA interface`	The probe has lost its connection to the CORBA.	The connection between the SpectroSERVER and the CA Aprisma devices is lost.
`Probe has lost connection to CORBA interface. Error internal to CORBA interface`	The probe has lost its connection to the CORBA.	The connection between the probe and the server is lost due to incorrect probe settings or inactive SpectroSERVER.
`New successful connection to the CORBA interface, connected to the Primary SpectroSERVER`	The probe connected to the primary SpectroSERVER.	The connection to the primary SpectroSERVER was successful.
`New successful connection to the CORBA interface, connected to the Secondary SpectroSERVER`	The probe connected to the secondary SpectroSERVER.	The primary SpectroSERVER has failed, and the probe has failed over the data to the secondary SpectroSERVER.

Appendix. Notices and Trademarks

This appendix contains the following sections:

- Notices
- Trademarks

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:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, 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
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
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 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.

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

IBM, the IBM logo, ibm.com, AIX, Tivoli, zSeries, and Netcool are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Adobe, Acrobat, Portable Document Format (PDF), PostScript, and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

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 are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Printed in USA

SC23-8593-01