IBM® Tivoli® Netcool/OMNIbus Probe for Huawei M2000 MML
3.0

Reference Guide
March 31, 2011

SC23-7870-01

Note

Before using this information and the product it supports, read the information in Appendix A, “Notices and Trademarks,” on page 15.

Edition notice

This edition applies to version 3.0.5930 of IBM Tivoli Netcool/OMNIbus Probe for Huawei M2000 MML (SC23-7870-01) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC23-7870-00.

Contents

Document control page

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

The IBM Tivoli Netcool/OMNIbus Probe for Huawei M2000 MML documentation is provided in softcopy format only. To obtain the most recent version, visit the IBM® Tivoli® Information Center:

https://www.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/common/Probes.html

Table 1. Document modification history

Document version	Publication date	Comments
SC23-7870-00	December 31, 2008	Summary table updated. IPv6 support information added. FIPS information added. Installation section added.
SC23-7870-01	March 31, 2011	Installation section replaced by “Installing probes” on page 2.

Chapter 1. Probe for Huawei M2000 MML

The Huawei M2000 system is a sub-network level management system that provides both element management and network management functionality for telecommunications networks. It communicates with external applications using a Man-Machine Language (MML) interface.

The Probe for Huawei M2000 MML connects to the MML interface and reads alarms from the Huawei M2000 system.

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 Huawei M2000 MML.

Table 2. Summary

Probe target	Huawei M2000 MML
Probe executable name	`nco_p_huawei_M2000_mml`
Package version	3.0
Probe supported on	For details of supported operating systems, see the following Release Notice on the IBM Software Support Website: https://www-304.ibm.com/support/docview.wss?uid=swg21414848
Properties file	`$OMNIHOME/probes/`*`arch`*`/huawei_M2000_mml.props`
Rules file	`$OMNIHOME/probes/`*`arch`*`/huawei_M2000_mml.rules`
Requirements	A currently supported version of IBM Tivoli Netcool/OMNIbus `probe-compatibility-3.x` (for IBM Tivoli Netcool/OMNIbus version 3.6) `common-libnetcool-3_1` (for IBM Tivoli Netcool/OMNIbus version 3.6) `common-libOpl-7_6` (for IBM Tivoli Netcool/OMNIbus version 3.6)

Table 2. Summary (continued)	
Connection method	TCP/IP
Remote connectivity	The Probe for Huawei M2000 MML can connect to a device on a remote host. Details of the remote host are specified using the **Host** and **Port** properties in the properties file.
Multicultural support	Available
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.
Federal Information Processing 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*.

Installing probes

All probes are installed in a similar way. The process involves downloading the appropriate installation package for your operating system, installing the appropriate files for the version of Netcool/OMNIbus that you are running, and configuring the probe to suit your environment.

The installation process consists of the following steps:

1. Downloading the installation package for the probe from the Passport Advantage Online website.

   Each probe has a single installation package for each operating system supported. For details about how to locate and download the installation package for your operating system, visit the following page on the IBM Tivoli Knowledge Center:

   http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/all_probes/wip/reference/install_download_intro.html
2. Installing the probe using the installation package.

   The installation package contains the appropriate files for all supported versions of Netcool/OMNIbus. For details about how to install the probe to run with your version of Netcool/OMNIbus, visit the following page on the IBM Tivoli Knowledge Center:

   http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/all_probes/wip/reference/install_install_intro.html
3. Configuring the probe.

   This guide contains details of the essential configuration required to run this probe. It combines topics that are common to all probes and topics that are peculiar to this probe. For details about additional configuration that is common to all probes, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

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
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.

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 Huawei M2000 MML acquires data from the Huawei M2000 system by connecting to the MML interface. It performs the chatin sequence specified by the **ChatinString** property and reads alarms generated by the Huawei M2000 system. It then parses the events and sends them to the ObjectServer.

Note : If an error occurs when the probe is reading from the socket, it discards the event and disconnects from the socket.

Data acquisition is described in the following topics:

Chat in and chat out scripts

Chat in and chat out scripts control probe login and logout. These scripts are on a single line in the expect-send format (for chat in scripts) or send-expect format (for chat out scripts). You can specify chat in and chat out strings using the **ChatinString** and **ChatoutString** properties or the -chatinstring and -chatoutstring command line options.

The format is:

```
ChatinString  : expect send expect send....
```

```
ChatoutString : send expect send expect....
```

Note : Each element in the chat strings is separated by white space. In order to send or expect a sequence that includes white space, surround the sequence with single quotes.

A typical chat in script might be:

```
.*login.*:.* anu\r\n .*assword.*:.* anu\r\n
```

The expect text can use any regular expression, while the send text can send any characters, including control characters using the standard UNIX/C escape sequences described in "Escape codes" on page 9.

Backoff strategy

If the **Retry** property is set to true, and the probe fails to establish a connection or loses an existing connection to the device, the probe reverts to a backoff strategy. The probe tries to reestablish a

connection after one second, two seconds, then four seconds, eight seconds, and so on, up to a maximum of 4096 seconds.

After the connection is made to the specified port, the probe tries to log in to the device. If the probe fails to log in, it shuts down and tries to connect again. The backoff strategy remains in place until a successful login occurs. The user can also specify a reconnection interval using the **ReconnectionInterval** property or `-reconnectioninterval` command line option. When this property is enabled, the probe reconnects at the specified time interval instead of using the backoff strategy.

If the remote host terminates the connection, the probe closes the connection on the host machine. The operating system is not allowed to close the connection.

Data stream capture

The probe can capture the data stream sent from a device in a stream capture file. For each connection, the full data stream is stored in a stream capture file

The data stream is stored using the following naming convention:

streamcapturefile_host_port

where:

- *streamcapturefile* is the value specified by the **StreamCaptureFile** property
- *host* is the name of the host to which the probe is connected
- *port* is the port on which the probe is listening for data

Stream capture data can be used for debugging purposes, to develop new features for the probe, or to pass to other management systems that require the same data.

Note : If you leave the **StreamCaptureFile** property blank, the data stream capture function is disabled.

Rotating stream capture files

The probe can rotate stream capture files; that is, it can write to a stream capture file that is saved and archived periodically when a predefined file size is reached

To use this feature, set the maximum size for the stream capture file using the **MaxStreamCapFileSize** property and set the **DateStreamCapture** property to 1.

When the **DateStreamCapture** property is set to 1, the probe creates a stream capture file with the following naming convention:

streamcapturefile_host_port_date_time

By appending the filename with a timestamp, the probe avoids overwriting the old stream capture file.

The maximum file size specified by the **MaxStreamCapFileSize** property acts as an upper limit for the stream capture file. If the probe reads a stream whose size exceeds the remaining allowable space in the current stream capture file, it saves the current file and creates a new one, thus storing the whole stream in a single file.

If you do not specify a maximum size for the stream capture file, it grows indefinitely until the connection is closed. If the **MaxStreamCapFileSize** property is set and the **DateStreamCapture** property is set to 0, the probe overwrites the stream capture file for that connection each time the maximum file size is reached.

ReadTimeout

The **ReadTimeout** property specifies how long the probe waits to read alarm data before timing out. Each time the probe attempts to read an alarm, this is the allotted time that it waits to receive data. If nothing is received, the probe moves on to the next alarm.

Inactivity

The probe has a timeout facility that allows it to disconnect from the socket if it fails to receive the next alarm data within a predefined amount of time. To specify how long the probe waits before disconnecting, use the **Inactivity** property. After this length of time, the probe disconnects from the switch, sends a ProbeWatch message to the ObjectServer, and tries to reconnect.

You can also use the **InactivityRetry** property to specify the number of consecutive inactivity intervals that the probe allows before disconnecting. If this property is set to 0, the probe disconnects after a single period of inactivity.

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"
```

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* (SC14-7530).

Table 4. Properties and command line options

Property name	Command line option	Description
ChatinString *string*	`-chatinstring` *string*	Use this property to specify the chat in script for connection to the host system. The default is `""`. A typical chat in script might be: `".*login.*:.*anu\r \n.*assword.*: .*anu\r\n".`
ChatoutString *string*	`-chatoutstring` *string*	Use this property to specify the chat out script for disconnection from the host system. The default is `""`.
DateStreamCapture *integer*	`-datestreamcapture` *integer*	Use this property to specify whether the probe appends a date and time to the stream capture filename: `0`: The probe does not append the date and time to the stream capture filename. `1`: The probe appends the date and time to the stream capture filename The default is `0`. **Note :** Setting the **DateStreamCapture** property to 1 prevents the stream capture file from being overwritten.
ExpectTimeout *integer*	`-expecttimeout` *integer*	Use this property to specify the time (in seconds) the probe allows for a response to the expect part of the chat in/out script. The default is `20`.

Table 4. Properties and command line options (continued)		
Property name	**Command line option**	**Description**
FlushTime *integer*	`-flushtime` *integer*	Use this property to specify the time (in seconds) the probe waits to receive the event terminator before sending the event to the ObjectServer. The default is `0`. **Note :** If this property is set to `0`, the probe only sends events to the ObjectServer on receipt of an event terminator.
Host *string*	`-host` *string*	Use this property to specify the name of the host to which the probe connects. The default is `localhost`.
Inactivity *integer*	`-inactivity` *integer*	Use this property to specify the time (in seconds) the probe allows a port to be inactive before sending the inactivity related ProbeWatch message. The default is `0` (probe does not disconnect if the port becomes inactive).
InactivityRetry *integer*	`-inactivityretry` *integer*	Use this property to specify the number of consecutive periods of inactivity the probe allows before attempting to reconnect to the host. The default is `0` (this feature is disabled).
MaxStreamCapFileSize *integer*	`-maxstreamcapfilesize` *integer*	Use this property to specify the maximum size (in bytes) of the stream capture file. When this limit is reached, the probe creates a new file. The default is `0`. **Note :** If the probe reads a stream whose size exceeds the remaining allowable space in the file, it creates a new one, thus storing the whole stream in a single file.
Port *integer*	`-port` *integer*	Use this property to specify the port to which the probe connects. The default is 8765.

Table 4. Properties and command line options (continued)		
Property name	**Command line option**	**Description**
ReadTimeout *integer*	`-readtimeout` *integer*	Use this property to specify the time (in milliseconds) the probe waits to read an alarm data before timing out. The default is `70000`. **Note :** The value set for the **ReadTimeout** property (in milliseconds) divided by 1000 should be less then the value set for **Inactivity** property (in seconds).
ReconnectionAttempts *integer*	`-reconnectionattempts` *integer*	Use this property to specify the maximum number of times that the probe attempts to reconnect to the socket. The default is `0` (probe makes unlimited attempts to reconnect to the socket).
ReconnectionInterval *integer*	`--reconnectioninterval` *integer*	Use this property to specify the ime (in seconds) between successive reconnection attempts. The default is `0` (probe uses standard backoff strategy).
StreamCaptureFile *string*	`-streamcapturefile` *string*	Use this property to specify the file the probe uses to store the input data stream. The default is `""`. **Note :** Leaving this property blank disables the stream capture function. When you no longer require data for debugging, you should disable the stream capture function.
TimeFormat *string*	`-timeformat` *string*	Use this property to specify the format used for the time stamp. The default is `%Y-%m-%d %H:%M:%S`.

Escape codes

You can use C-style escape codes in the **ChatinString** and **ChatoutString** properties. This allows you to easily define whether to send escape code sequences after commands.

For example, the following chatin string sends a carriage return character (`\r`) after the user name and password:

```
ChatinString : ".*: user\r .*: passwd\r .*:"
```

The following table explains the character sequences that are recognized.

Table 5. ChatinString escape codes	
Escape code	**Character**
\b	This escape code specifies the backspace character.
\f	This escape code specifies the form-feed character.
\n	This escape code specifies the new-line character.
\r	This escape code specifies the carriage return character.
\t	This escape code specifies the tab character.
\\	This escape code specifies the backslash character.
\'	This escape code specifies the single quote character.
\"	This escape code specifies the double quote character.

Note : Due to the way in which the above properties are parsed, the escape sequences for backslash, single quote and double quote must be double-escaped. For example, to send a backslash character (\), use \\\\.

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 Huawei M2000 MML generates. Not all the elements described are generated for each event; the elements that the probe generates depends upon the event type.

Table 6. Elements	
Element name	**Element description**
$AckTime	This element displays the time at which the alarm was acknowledged.
$AlarmID	This element displays the identifier of the alarm.
$AlarmName	This element displays the name of the alarm.
$Category	This element displays the category of the alarm.
$CategoryID	This element displays the identifier of the alarm category.

Table 6. Elements (continued)	
Element name	**Element description**
$ClearTime	This element displays the time at which the alarm was cleared.
$EventID	This element displays the identifier of the alarm type.
$EventType	This element displays the type of alarm.
$Location	This element displays the location information of the alarm.
$NeFdn	This element contains the identifiers of the network elements.
$NeName	This element displays the name of network element.
$NeSn	This element displays the Alarm Serial Number.
$NeType	This element displays the type of network element.
$ObjFdn	This element contains the identifiers of the object.
$ObjName	This element displays the name of the object.
$ObjType	This element indicates the type of the object.
$OccurTime	This element displays the time at which the alarm occurred.
$Operator	This element displays the operator who acknowledged the alarm.
$Severity	This element displays the severity of the alarm.
$SeverityID	This element displays the identifier of the alarm severity.
$Sn	This element contains the Equipment Serial Number of the alarm.
$State	This element displays the state of the alarm.

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* (SC14-7530).

Table 7. Error messages

Error	Description	Action

`Could not remove leading and trailing spaces from element`
`Failed to initialize buffer`
`Failed to initialize line buffer`
`Failed to remove leading and trailing spaces from element`
	This is an internal error.	Make more memory available. If the problem persists, refer to your support contract for more information about contacting the help desk.

`Failed to read from socket`	This is an internal error.	Check the connectivity between the probe and the device.
`Date stamp not in recognized format`	The property value received did not match the date stamp format expected.	Ensure that the date stamp adheres to the format specified.
`Failed to connect to ObjectServer - aborting`	The ObjectServer appears to be unavailable.	Check that the ObjectServer is running. Check that the interfaces file on the system where the probe is installed has an entry for it. Check that there is no networking problem between the Huawei server and the ObjectServer.
`Failed to read from socket - disconnecting`	There is a problem with your network or Huawei M2000 MML.	Check the network connectivity between the probe and the Huawei M2000 MML interface.
`Failed to send 'Running ...' ProbeWatch` *`message reason`*	The probe failed to send the ProbeWatch message to the ObjectServer.	Check the network connectivity and check that the ObjectServer is running correctly.
`Failed to translate date stamp into UTC`	The probe was unable to translate the TIME token into UTC format. There may be a problem with the format of the alarm; the incoming data stream may be corrupt.	Check that the Huawei M2000 MML is running correctly. Check the connection to the MML interface.
`Memory allocation error`	The probe has insufficient memory.	Make more memory available.
`Message is too long for buffer - truncating`	Unable to allocate memory for the buffer that contains the event being read. This caused the probe to terminate.	Refer to your support contract for more information about contacting the help desk.

Table 7. Error messages (continued)		
Error	**Description**	**Action**
`Socket read timeout has occurred`	The probe could not read from the socket within the time specified by the **ReadTimeout** property.	Check the network connection. If the connectivity is slow, increase the **ReadTimeout** property value.

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* (SC14-7530).

Table 8. ProbeWatch messages		
ProbeWatch message	**Description**	**Triggers/causes**
`Disconnected from system due to shutdown signal`	The probe disconnected from the system.	The probe received a shutdown signal.
`System is inactive.. host,port number`	The server is currently inactive.	The system has been inactive for a period greater than that specified by the Inactivity property.
`Disconnecting from system due to Inactivity... host, port number`	The probe closed the connection to the Huawei server.	The server was inactive for the number of consecutive inactivity periods specified by the **InactivityRetry** property.
`Connection lost ...host,port number/`	The connection has been lost.	The probe lost the TCP/IP connection to the Huawei server.
`Going Down ...`	The probe is shutting down.	A shutdown signal was sent to the probe.
`Failed to get events`	The probe failed to get any events.	The probe could not get any events from the Huawei server.
`Running ...`	Probe is running	The probe started and is running normally.

Appendix A. 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.

SC23-7870-01