IBM® DB2® for Linux®, UNIX®, and Windows®

Setting DB2 Client Information Fields in Common Middleware Applications

Supplement to Best Practices: Implementing DB2 Workload Management

Paul Bird

Senior Technical Staff Member
DB2 Development
IBM Toronto Laboratory

Issued: December 2012

Introduction

The DB2® client information fields are available on each connection to a database. These fields enable an external application that is using a connection to provide additional information to the DB2 database server that can be used to discriminate among connections based on end-user identification. The values in the client information fields are reported by DB2 for Linux®, UNIX®, and Windows® and other members of the DB2 family through various database monitoring and auditing interfaces. They are also leveraged by the DB2 workload definition in DB2 for Linux, UNIX, and Windows Version 9.5 and later as another way to aggregate connections to the database for purposes of monitoring and control.

There are four DB2 client information fields:

- Client Acctng (SQL_ATTR_INFO_ACCTSTR)

  (http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.mon.doc/doc/r0055510.html)

- Client Applname (SQL_ATTR_INFO_APPLNAME)

  (http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.mon.doc/doc/r0055512.html)

- Client Userid (SQL_ATTR_INFO_USERID)

  (http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.mon.doc/doc/r0055511.html)

- Client Wrkstnname (SQL_ATTR_INFO_WRKSTNNAME)

  (http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.mon.doc/doc/r0055513.html)

For further information about these fields, see the DB2 documentation.

This document includes the latest information about a select number of popular middleware applications that are known to provide either a manual or automatic way to set these values.

Although every attempt has been made to validate the contents of this document, the information in this document should be treated as an initial suggestion. For more detailed or complete information about a specific middleware application, see the official documentation for that application.

Special Note on the CLIENT PROGRAMNAME Attribute

There is another information field that can be set for a connection, but only prior to the establishment of that connection: PROGRAMNAME or CLIENT PROGRAMNAME. Setting this field overrides the default application name that is used by the DB2 database manager for the connection (and that is returned by the **appl_name** monitor element). For more information about this field, see http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.mon.doc/doc/r0001165.html.

Depending on the DB2 client that is being used in a given environment, there might be a mechanism to provide a value for the SQL_ATTR_INFO_PROGRAMNAME attribute with a CONNECT request. Consult the most recent client documentation for accurate information. For examples of such a mechanism, see http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.swg.im.dbclient.config.doc/doc/r0054662.html.

General Techniques for Setting Client Information Fields

There are three ways to change the value of client information fields:

- The **sqleseti** administrative API interface
- The WLM_SET_CLIENT_INFO stored procedure
- DB2 client mechanisms

With each mechanism, it is important to note that DB2 workload management will not recognize any change in these values (in terms of assignment of the connection to a DB2 workload) until the beginning of the next unit of work. However, any change to the client information values will be reflected immediately in the associated special registers and monitoring interfaces.

The **sqleseti** administrative API is an interface that any external application can invoke in its source code. For more information, see http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.apdv.api.doc/doc/r0001709.html.

The WLM_SET_CLIENT_INFO stored procedure is a built-in DB2 procedure that is provided with the database server. For more information, see http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.sql.rtn.doc/doc/r0053116.html.

Each DB2 client typically provides a way for an application to set the client information values by invoking some local client interface. Different interfaces are used for different clients.

For CLI clients, call the SQLSetConnectAttr() function by specifying values for one or more of the following connection attributes:

- SQL_ATTR_INFO_WRKSTNNAME
- SQL_ATTR_INFO_ACCTSTR
- SQL_ATTR_INFO_APPLNAME
- SQL_ATTR_INFO_USERID

For more information, see http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.apdv.cli.doc/doc/r0006816.html.

For IBM Data Server Driver for JDBC and SQLJ versions before Version 4.0, IBM provides extensions to DB2Connection that you can use to set the client information fields. For more information, see http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.apdv.java.doc/src/tpc/imjcc_tjvstcli.html.

For IBM Data Server Driver for JDBC and SQLJ Version 4.0 or later, the DB2Connection extensions have been deprecated, and you are directed to use the java.sql.Connection.setClientInfo method. For more information, see http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.apdv.java.doc/src/tpc/imjcc_t0052428.html.

Products That Offer Automatic Settings

IBM Cognos® Business Intelligence

Cognos Business Intelligence 8 and 10 can be configured to automatically set the client information fields by using the WLM_SET_CLIENT_INFO stored procedure or by using the mechanism that is provided by the DB2 client. For more information, see http://publib.boulder.ibm.com/infocenter/cbi/v10r1m0/topic/com.ibm.swg.im.cognos.ug_cra.10.1.0.doc/ug_cra_id13715DB2ConAtt.html#DB2ConAtt.

Cognos Business Intelligence 8.4.1 Fix Pack 2 and later can be configured to use the DB2 client mechanism to provide DB2 values for these fields, which are then used for recognition by and integration with IBM Optim™ Performance Manager (OPM). This facility is available for connections that are defined to use the CLI interface in the Data Server client. This configuration can also be used with DB2 servers where workload management rules might need to be established at a finer level of granularity. When the DB2 client mechanism is used, Cognos Business Intelligence will automatically update the DB2 client settings prior to preparing a statement. The client settings are packed in a specific format, as described in the Cognos Business Intelligence documentation.

For information about the fields and slots that are filled in automatically, see the Cognos documentation (http://publib.boulder.ibm.com/infocenter/cbi/v10r1m0/topic/com.ibm.swg.im.cognos.ug_cra.10.1.0.doc/ug_cra_id13715DB2ConAtt.html#DB2ConAtt).

Because Cognos is indifferent to the DB2 product being used when setting these fields, the most stringent restrictions on size and inputs apply to all platforms. And because the field limits are expressed in bytes, not characters, Cognos attempts to detect when a character would require more than one byte and emits a single character to avoid length problems when using Unicode encoding; an attempt is made to "compress" the data to "fit" without discarding too much context.

In general, the hierarchy of folders in your Cognos content store, coupled with the size of object names, has a great impact on whether the resulting values will fit in the available space. To avoid overflowing the available space in these fields, consider simplifying your Cognos organization when using a tool like OPM.

IBM WebSphere® Application Server

WebSphere Application Server Version 6.0 (and later) can be configured to automatically set the client information fields for a connection to a DB2 data server. This is accomplished by enabling the WebSphere Application Server trace group WAS.clientinfo for each application server. For more information, see http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.nd.multiplatform.doc/ae/rdat_impclientinfo.html.

The following fields and slots are filled in automatically. For more detailed information, see the WebSphere documentation.

- **Client Userid**. The name of the user that initiates the application request. When supported, this option is collected and passed to the database back end. This information is collected by calling the WSSecurityHelper.getFirstCaller method.

- **Client Wrkstnname**. The name of the user's location, in the form of cell:node:server. This information is collected by calling the WSSecurityHelper.getFirstServer method.

- **Client Applname**. The name of the running application. This value is returned from the J2EEName object by the getApplication method.

Important: The Client Userid and Client Wrkstnname values can be automatically set on a database connection only if you enable Java 2 security. The Client Applname value is collected regardless of the security setting.

IBM InfoSphere® DataStage®

Starting with IBM InfoSphere Information Server Version 8.5 and later, the DataStage product automatically sets the DB2 client information fields when a DB2 connector stage is used within a job. For more information about the DB2 connector, see http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r5/index.jsp?topic=%2Fcom.ibm.swg.im.iis.conn.db2db.usage.doc%2Ftopics%2Fdb2_connector_top_of_nav.html.

The following fields and slots are filled in automatically:

- **Client Userid**. The name of the user that is running the job.

- **Client Wrkstnname**. The unqualified host name of the DataStage server.

- **Client Applname**. Identifies the DataStage project and job. The format of the string is: `<project name> || ':' || <job name>`. If this string is longer than 32 characters, it flows over to $SLOT1 in the accounting string.

- **Client Acctng**. The format of the string is: `'DS ' 'ccc' 'vr' $slot1 $slot2 $slot3 $slot4`

    - The prefix 'DS ' associates the request with IBM DataStage products in IBM Optim Integrated Data Management.

    - 'ccc' represents three characters that identify the mode under which DataStage is executing this job. Valid values are 'Par' (for parallel) and 'Ser' (for server).

    - 'vr' identifies the DataStage version and release, and will be '85' for the first release that supports this new functionality.

Additional accounting information is divided into four fields:

- $SLOT1 (overflow area for SQL_ATTR_INFO_APPLNAME, truncated if total length is > 78 characters) = <project name> || ':' || <job name>
- $SLOT2 = 'Partition ' || <partition number> || ' of ' || <partition count>
- $SLOT3 = <invocation ID>
- $SLOT4 = <stage name>

Each slot has a fixed length (maximum 46) and is padded with blanks if necessary. Longer strings are truncated.

SAP Business Information Warehouse on DB2 for z/OS®

SAP Business Information Warehouse 7.0 SP22 on DB2 for z/OS (and later) can automatically provide values for the DB2 client information fields. This is enabled when the dbs/db2/use_accounting profile parameter is set to 1. For more information, see SAP note 757329 (https://service.sap.com/sap/support/notes/0000757329?nlang=E)[1].

After the profile parameter is set to 1, the following fields are filled in automatically:

- **Client Userid**. The name of the SAP end user.
- **Client Wrkstnname**. For batch processes, provides the SAP batch job name.
- **Client Applname**. For batch processes, provides the ABAP report name. For dialog processes, provides the InfoProvider or BW query name.

SAP Application Server ABAP on DB2 for Linux, UNIX, and Windows

SAP Application Server ABAP Version 7.0 and later automatically provides values for the DB2 client information fields. For more information, see SAP note 1059905 (https://service.sap.com/sap/support/notes/0001059905?nlang=E)[1].

The following fields are filled in automatically:

- **Client Userid**. The name of the SAP user being logged on to the SAP application.
- **Client Wrkstnname**. The application server name.
- **Client Applname**. The SAP work process type (for example, 'ptype UPD', 'ptype DIA'). If the SAP transaction code is known in a dialog work process, the transaction code is written instead of 'ptype DIA'.
- **Client Acctng**. The ABAP report or BW query name.

[1] A SAP Service Marketplace authentication is required to access this information.

The SAP Application Server also ensures that the **SESSION_AUTH_ID** special register is set to the value of the database connection ID, which is typically 'sapr3' or 'sap<sapsid>', where <sapsid> = 3*[0-9 a-z].

SAP Application Server Java on DB2 for Linux, UNIX, and Windows

SAP Application Server (AS) Java Versions 7.02, 7.30, and later automatically provide values for the DB2 client information fields. For more information, see SAP note 1569638 (https://service.sap.com/sap/support/notes/0001569638?nlang=E)[1] .

The following fields are filled in automatically:

- **Client Userid**. The name of the SAP AS Java engine user.
- **Client Wrkstnname**. The application server name, determined by the DB2 data server from connection attributes.
- **Client Applname**. The name of the AS Java engine thread that is currently using this database connection. The name of the thread is derived with the help of the java.lang.Thread.currentThread() method. For SAP NetWeaver releases 7.30 and later, the native thread ID, which is acquired by using the java.lang.Thread.currentThread.getId() method, is concatenated to the thread name with a colon (:) that serves as a delimiter between the thread name and the thread ID.
- **Client Acctng**. Always set to 'SAP J2E'.

The SAP Application Server also ensures that the **SESSION_AUTH_ID** special register is set to the value of the database connection ID, which is typically 'sapr3db' or 'sap<sapsid>db', where <sapsid> = 3*[0-9 a-z].

Products That Offer Manual Settings

IBM Cognos Business Intelligence

IBM Cognos Business Intelligence 8 and 10 can be configured to set client information by calling the WLM_SET_CLIENT_INFO stored procedure or by using DB2 client mechanisms. For more information, see http://publib.boulder.ibm.com/infocenter/cbi/v10r1m0/topic/com.ibm.swg.im.cognos.ug_cra.10.1.0.doc/ug_cra_id13715DB2ConAtt.html#DB2ConAtt.

Cognos Business Intelligence administrators can configure database connections to execute a series of statements against a DB2 server while reports are being executed. Administrators can customize what values will be passed to the WLM_SET_CLIENT_INFO stored procedure.

For DB2 connections, the WLM_SET_CLIENT_INFO stored procedure can be invoked from within a Cognos command block, as shown in this example: http://publib.boulder.ibm.com/infocenter/cbi/v10r1m0/topic/com.ibm.swg.im.cognos.ug_cra.10.1.0.doc/ug_cra_id13581asg_example_passing_auth_provider_details.html#asg_example_passing_auth_provider_details.

If you want to track usage by coarsely-grained concepts such as an "application" (for example, HR reporting or Finance reporting) or business entities (for example, the business unit to which a user belongs, or a geographical region), then you should use the manual approach to call the WLM_SET_CLIENT_INFO stored procedure. To track incoming work at the granularity of a report, or a named query within a report, use the automatic setting that is offered through the DB2 client mechanisms.

Note that the manual setting of client information fields might interfere with how Cognos Business Intelligence is recognized and reported by the IBM Optim Performance Manager, because that product relies on the setting of well-defined values (noted previously in "IBM Cognos® Business Intelligence") to automatically detect the existence of the Cognos Business Intelligence product.

IBM WebSphere® Application Server

WebSphere Application Server Version 6.0 (and later) also provides a manual way to explicitly set client information values for connection to a DB2 database. This is best described in the WebSphere documentation (http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.nd.multiplatform.doc/ae/tdat_clientinfotask.html).

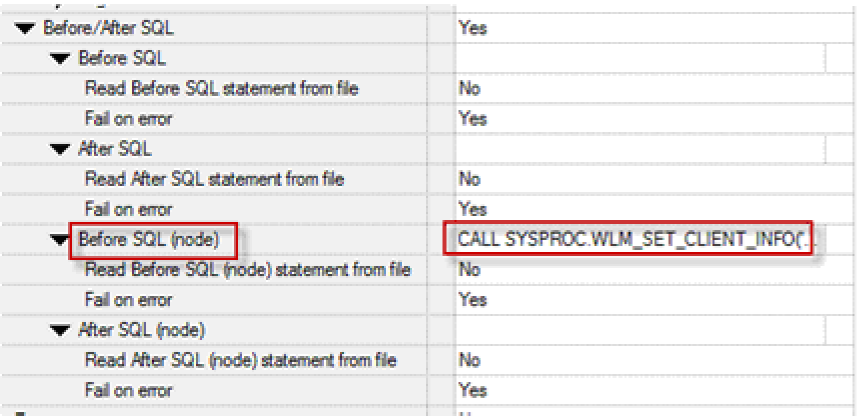
Note that the manual setting of client information fields might interfere with how WebSphere Application Server is recognized and reported by IBM's Optim Performance Manager, because that product relies on the setting of well-defined values (noted previously in "WebSphere Application Server") to automatically detect the existence of the WebSphere Application Server product.

IBM InfoSphere DataStage

The IBM InfoSphere DataStage product also provides different ways to set the client information fields manually. Using a CALL SQL statement, simply inject an invocation of the WLM_SET_CLIENT_INFO stored procedure into the SQL that is being executed against a DB2 database server.

One way to add such an invocation without impacting the actual SQL content for a stage is to leverage the DB2 Connector's "Before SQL (node)" property to make a call to the WLM_SET_CLIENT_INFO procedure prior to the SQL that was identified for the stage being executed.

The following figure shows an example of how this might be done. The editor opens when you double click a stage in a job design.



▼ Before/After SQL	Yes
▼ Before SQL	
Read Before SQL statement from file	No
Fail on error	Yes
▼ After SQL	
Read After SQL statement from file	No
Fail on error	Yes
▼ Before SQL (node)	CALL SYSPROC.WLM_SET_CLIENT_INFO(
Read Before SQL (node) statement from file	No
Fail on error	Yes
▼ After SQL (node)	
Read After SQL (node) statement from file	No
Fail on error	Yes

Figure 1: Setting client information fields with InfoSphere DataStage

Note that the manual setting of client information fields might interfere with how InfoSphere DataStage is recognized and reported by IBM's Optim Performance Manager, because that product relies on the setting of well-defined values (noted previously in "DataStage") to automatically detect the existence of the InfoSphere DataStage product.

MicroStrategy

MicroStrategy applications provide a way to reference internal MicroStrategy context and to use this to explicitly set client information values for a connection to a DB2 database.

The basic concepts and approach using DB2 9.5 for Linux, UNIX, and Windows are described in a joint MicroStrategy and IBM white paper (http://public.dhe.ibm.com/software/data/sw-library/db2/papers/wlm_msi_db29.5.pdf). For the most recent information, see the relevant MicroStrategy documentation.

SAP BusinessObjects XI 3.1 and Later

SAP BusinessObjects XI 3.1 (and later) provides a manual way to explicitly set client information values for a connection to a DB2 database. Two approaches are described in the BusinessObjects XI documentation, but the following brief summary is included here to provide some guidance. Note that neither approach has been explicitly tested by the author or by IBM, and that adjustments might be required to ensure a successful implementation. The following information is provided as a suggestion to help you integrate BusinessObjects and the DB2 product, and it is strongly recommended that you engage with BusinessObjects technical support if any issues arise.

In the case of DB2 workload management, the objective is to inject a call into the WLM_SET_CLIENT_INFO stored procedure *and* a COMMIT statement so that the connection can be evaluated for potential association with a new DB2 workload definition, which occurs only at the start of each new unit of work. To achieve this objective, you want to be able to execute an SQL invocation that is similar to the following example:

```
CALL WLM_SET_CLIENT_INFO('toto',  'kansas', NULL, NULL, NULL);

COMMIT;
```

The first suggested technique involves the BEGIN_SQL function, which is documented as allowing you to inject SQL prior to the execution of BusinessObjects-generated SQL statements. A description of this technique can be found near the bottom of the information that is available at this site: http://e-university.wisdomjobs.com/sap-bo/chapter-757-252/optimizing-universes.html. The BEGIN_SQL function is processed at the universe level, and so should work well with WebI reports, as well as both CLI and JDBC connections.

If you are using this approach, note that the BEGIN_SQL parameter is processed before each SQL statement is issued by BusinessObjects, which might have the unwanted effect of separating shared SQL into different DB2 transactions (because of the COMMIT statement that is appended after the CALL statement). One other possible restriction noted in public sources is that the BEGIN_SQL parameter might be ignored by Desktop Intelligence and Crystal Reports. Again, it is best to discuss this potential issue with BusinessObjects technical support if it is deemed to be relevant to your environment.

An alternate approach for setting the DB2 client information fields is to use the ConnectInit parameter, which enables you to specify an SQL statement to be executed as part of the base connection initialization. This parameter is processed at the connection level, and so might work in work flows that do not involve WebI reports. Note that its use is restricted to CLI connections.

BusinessObjects also offers a number of system and report variables that can be referenced in the SQL that is being injected through the parameters that are described above. The values of these variables are accessed by using the @Variable function; for example, in your BEGIN_SQL invocation, you could use the following invocation to pass in the current BusinessObjects user login ID to the DB2 data server:

```
CALL WLM_SET_CLIENT_INFO(@Variable('BOUSER'), NULL, NULL, NULL,
NULL)
```

Note that not all system variables are available for use in the ConnectInit parameter, because not all variables are available when a connection is established.

The following list includes some of the variables that might be of interest:

- BOUSER. BusinessObjects user login ID.
- DOCNAME. Name of the current document.
- DPNAME. Name of the data provider.
- DPTYPE. Type of the data provider.
- UNVNAME. Name of the universe.

- UNVID. ID of the universe.

SAS/ACCESS 9.2 and Later

It is believed that SAS/ACCESS 9.2 (and later) provides a manual way to explicitly set client information values for a connection to a DB2 database. Note that neither approach has been explicitly tested by the author or by IBM, and that we have been unable to completely validate this information with SAS. The following information describes a suggested approach to integrating SAS/ACCESS and the DB2 product, but it is not guaranteed to be successful. It is strongly recommended that you engage with SAS technical support to confirm the validity of the suggested approach and to seek help with any issues that might arise.

The suggested technique uses the LIBNAME option DBCONINIT to inject a call to set the client information fields on each connection when it is established. A description of this technique can be found at this site: http://support.sas.com/kb/13/147.html.

In the case of DB2 workload management, the idea is to inject a call into the WLM_SET_CLIENT_INFO stored procedure *with* a COMMIT statement so that the connection can be evaluated for potential association with a new DB2 workload definition, which occurs only at the start of each new unit of work.

Note the following caveats if you are using the DBCONINIT approach:

- Only a single SQL statement or procedure can be specified with this option. It is unknown whether a commit occurs automatically as part of SAS/ACCESS processing, or not; if not, a “wrapper” stored procedure must be created to do both the CALL statement and the COMMIT statement.

- The initialization command might execute more than once, because one LIBNAME statement might have multiple connections, such as one for reading and one for updating. The net effect should be the same, because the read and update work (for example) will be going through the same workload definition, even though they are running on different connections and will appear as different workload occurrences for that same workload definition.

If the SAS jobs themselves can be modified and the procedure used does not return a result set of output parameters, then another possible approach for setting the DB2 client information fields is to use PROC SQL PASS-Through. A description of this technique can also be found at this site: http://support.sas.com/kb/13/147.html.

It is unknown whether one can use SAS automatic macro variables as input arguments for any injected call to the WLM_SET_CLIENT_INFO stored procedure with these approaches. Again, consultation with SAS technical support is advised.

If using these SAS automatic macro variables is possible, the following variables might be of interest:

- SYSUSERID. Contains the user ID or login name of the current SAS process.

- SYSPROCNAME. Contains the name of the procedure (or DATASTEP for DATA steps) that are currently being processed by the SAS Language Processor.

- SYSHOSTNAME. Contains the host name of the system that is running a single TCP/IP stack.

- SYSJOBID. Contains the name of the current batch job or user ID.

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.

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.

Without limiting the above disclaimers, IBM provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any recommendations or techniques herein is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Anyone attempting to adapt these techniques to their own environment do so at their own risk.

This document and the information contained herein may be used solely in connection with the IBM products discussed in this document.

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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

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.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml

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

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

Linux is a registered 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.

Contacting IBM

To provide feedback about this paper, write to db2docs@ca.ibm.com

To contact IBM in your country or region, check the IBM Directory of Worldwide Contacts at http://www.ibm.com/planetwide

To learn more about IBM Information Management products, go to http://www.ibm.com/software/data/