Best Practices
Upgrade InfoSphere MDM With Near-Zero Production Downtime

Wenjie Lin
Senior Technical Specialist
InfoSphere MDM Lab Services
IBM Software Group
Contents

Upgrade InfoSphere MDM With Near-Zero Production Downtime........1
Contents..................................................................................................................2
Executive Summary..............................................................................................3
Final Move to Production..................................................................................10
Summary..............................................................................................................22
Further reading...................................................................................................23
    Contributors.................................................................................................23
Notices..................................................................................................................24
    Trademarks..................................................................................................25
Executive Summary

Achieving an objective of near-zero production downtime in the upgrade of InfoSphere Master Data Management (InfoSphere MDM) presents a challenging task for the implementation team. It requires careful planning and execution. This article will introduce you to the common upgrade steps and strategies for achieving near-zero production downtime for InfoSphere MDM Advanced Edition (previously known as InfoSphere MDM Server).

Your upgrade strategy will depend on many factors, including:

• Making decisions about the new capabilities you’d like to leverage from the latest version.

• Migrating your existing customizations and handling deprecation of services or capabilities.

• Considering what the hardware and software upgrade will mean for existing consuming applications.

• Ensuring the migration plan can occur within a time frame that makes sense for your business.

The upgrade process includes a number of steps: the Base Installation Pass, Code Migration Pass, the Server Migration Pass, and the Final Move to Production Pass. Throughout this document, the activities to be performed in each of these passes are illustrated in detail and supplemented with examples. We also make the following recommendations to facilitate the migration process:

1. Isolate potential code migration issues from database upgrade and application deployment issues by testing migrated customizations with new MDM version as a separate upgrade activity.

2. Adopt new functional capabilities in a separate phase of the project, after completing the migration to the new version. For example, as of version 10.0, InfoSphere MDM provides an option to use a probabilistic matching engine (PME) for identifying suspects. If you want to switch to this new probabilistic matching capability, we recommend that this be done as a separate activity following the upgrade of your InfoSphere MDM solution with its existing capabilities remaining in place.

3. Use an incremental approach to migration that includes planning for and testing each step of the process, including Development (DEV), Integration Testing (IT), and User Acceptance Testing (UAT) of environments before moving to production servers.

4. Create and execute a thorough regression test plan that covers all business scenarios with all InfoSphere MDM client applications.

5. Use high-availability infrastructure and database synchronization tools in the upgrade of production servers.
Planning for Upgrade

With each new release of InfoSphere Master Data Management, many new capabilities are introduced. As such, there are many reasons to consider upgrading:

- An upgrade may provide access to new functionality that will better position you to meet new and existing business objectives.
- An upgrade may facilitate taking advantage of new technologies.
- An upgrade allows you to leverage the latest product performance and usability enhancements and may improve the efficiency of your InfoSphere MDM solution.
- You may wish to upgrade in order to move from a version of the product that is at end of life.

The process of migrating to a newer version of InfoSphere MDM should always start by defining what the objectives of the upgrade are and evaluating how to meet those objectives. As part of this evaluation, there are many factors to consider, such as your new business needs, product features and functionality that address those needs, dealing with any deprecated product capabilities, technical infrastructure changes, any changes needed to deployment topology, and potentially integration with other software to support your MDM solution. These factors are often interrelated. Considered systematically, each factor is important in determining whether to upgrade. If you do choose to upgrade, you must develop strategies to address each of these factors within the time frame for upgrade completion.

Once the decision has been made to upgrade, planning for the migration is the next step. A plan for an upgrade project is, in many respects, similar to a plan for an implementation project. It will involve a resource plan, infrastructure plan, new function test plan, regression test plan, performance test plan, and more. Upgrading InfoSphere MDM typically requires new infrastructure, such as a new version of the application server, database, and messaging technology. The required stack of infrastructure must be prepared prior to MDM Sever product installation.

We recommend planning for four activities, or passes, to cover the upgrade process: the Base Installation Pass, the Code Migration Pass, the Server Migration Pass, and the Final Move to Production Pass. Key to note is that each of these passes has an accompanying test plan to be executed along with them. Both functional and performance testing at each of these stages is very important to ensure you retain or achieve your service level agreements. Be sure to refer to our best practices articles that provide prescriptive guidance on both implementing for performance and performance tuning the InfoSphere MDM solution environment.

Figure 1 provides a high-level overview of the upgrade process for InfoSphere MDM. You will notice that it includes the following steps:

- Upgrade Assessment and Plan (covered in this section)
- Base Installation: make sure that your new version of InfoSphere MDM is installed and working correctly
• Code Migration: ensure all of your customizations work correctly with the new version

• Server Migration Pass

• Move to Production: move all of your client applications to the new version

---

**Figure 1: InfoSphere MDM Upgrade Process**

The remainder of this document focuses the procedures to be performed at each of these stages. These procedures are supplemented with examples.
Base Installation Pass

In the Base Installation Pass, the new product release is installed on servers with a new temporary database. For example, you can install the latest version of InfoSphere MDM with a temporary database for that version when upgrading from an earlier version of InfoSphere MDM. The temporary database will be used to verify the base InfoSphere MDM installation. The testing of your customizations happens in a later stage.

The latest product version is installed along with new versions of the client applications such as the MDM batch client, the MDM Message Adapter and product UIs. Configurations of existing MDM clients get incorporated into newly installed MDM clients, such as the numbers of the submitters in the MDM batch client or the number of the sessions in the MDM Message Adapter. Once the installation is complete, a thorough test of the base installation is performed to make sure all newly installed MDM clients are working correctly with the newly installed version of InfoSphere MDM. Once the new release and its associated client programs have been installed and verified, all available product fix packs are applied so that your installation is up to date.

Best Practices

- Keep current. Apply all available product fix packs once the base installation has been completed.
- Test your new installation thoroughly to verify all required components are working as expected.

Code Migration Pass

The Code Migration Pass can be executed in parallel with the Base Installation Pass. Most InfoSphere MDM implementations involve customizations, such as data extensions and additions, behavior extensions, composite transactions, code table additions, and custom external rules. In this stage of the upgrade process, you will ensure that all customizations created in an earlier version of InfoSphere MDM continue to work as expected with the new version of the software. The technology underpinning your customizations may have changed version-to-version. Any customizations created in the MDM Workbench can be migrated therein to take care of these differences for you.
Customization Identification
The first task in the Code Migration Pass is to gather current InfoSphere MDM customization information. The team performing the upgrade will need to identify the customizations that are deployed in the MDM solution. Frequently, upgrades are performed by a team other than the initial implementation team, so the upgrade team must understand the current state of the implementation’s customizations. To facilitate information gathering, the upgrade team will not only need the source code for the customizations, but also the functional requirements and technical design artifacts that supported this work.

Migration of Customizations
The MDM Workbench gets installed into IBM Rational Software Architect or IBM Rational Application Developer. By importing your new model into the latest version of the MDM Workbench and regenerating your customizations, any code incompatibilities between releases will be automatically resolved by the MDM Workbench tools. When upgrading from IBM WebSphere Customer Center (WCC) 6.5 or WCC 7.0.x to InfoSphere MDM 9.0.x or 10.x, it is recommended to regenerate MDM Extension/Addition codes using the more current InfoSphere MDM Workbench to leverage the new UML module model, and to upgrade to the most recent data persistency strategy (pureQuery instead of the CMP EJB).

When migrating to MDM v11, the MDM Workbench will automatically enable OSGi support for customizations generated from workbench model. Java customizations not supported by MDM Workbench would need to be migrated manually by following instructions in MDM Information Center.

Before regenerating your code, ensure that any manually implemented custom code in your customizations has the “NOT GENERATED” annotation to prevent it from being overwritten. For more information on code migration using the MDM Workbench, please consult the InfoSphere MDM Information Center.

After migrating your customizations in the Workbench, review your code and take note of any remaining references to deprecated classes or utility functions. Take the appropriate actions to resolve and remove the use of deprecated APIs.

It is recommended that the code migration and functional testing be performed on the development workstation to start. Be sure to test all custom code in the development environment as this will help to isolate any problems with the code from potential issues that may surface as part of the database upgrade or application deployment to the server environment.

Testing Customizations
As part of testing your code migration, run the original test suites from the previous implementation project. These are your regression tests. Running these tests will ensure that your customizations continue to function as designed. If you do not have an existing regression test suite, it should be created now. This regression test suite can be used in subsequent server migration tests. At the end of the Code Migration Pass, you should be able to incorporate all existing customizations into the new release of InfoSphere MDM, with acceptable test results.
Best Practices

- Migrate the code to your existing customizations and test thoroughly in the development environment to isolate code migration issues from application and database upgrade issues that may be uncovered as part of the server migration activity.

- Review your code and take the appropriate actions to resolve and remove the use of deprecated APIs (ex. utility classes) you may have introduced in your customizations.

- Add new customizations as a separate phase of the project, after your existing ones have been tested and work as designed.

Server Migration Pass

After the completion of the Base Installation Pass and the Code Migration Pass, the migrated customizations must be packaged into the new application archive for the upgraded InfoSphere MDM application. It must then be deployed and tested in a number of environments, starting from the DEV (development) test server, and then moving to the Integration Test (IT) and User Acceptance Test (UAT) servers. Again, in each of these environments, there should be performance testing done to ensure that your functional and non-functional goals are met. The objective of this pass is to ensure that the upgrade process on servers is complete and that all existing client applications work correctly with the upgraded InfoSphere MDM application. The completion of this pass will significantly reduce the time required for the production upgrade and minimize the risk associated with the production upgrade.

The following list provides a summary of the activities that should be performed in this pass:

**Step 1:** Run all InfoSphere MDM customization database scripts on the temporary database that was created in the Base Installation Pass. For example, you can execute InfoSphere MDM customization database scripts on the newly created temporary InfoSphere MDM 9.0.2 database when upgrading an earlier version of InfoSphere MDM to version 9.0.2.

**Step 2:** Prepare the InfoSphere MDM application. For InfoSphere MDM versions prior to version 11, this involves incorporating your customizations into the .ear file to be deployed on servers.
Deploy the InfoSphere MDM with customizations onto a server. At this point, the new release of InfoSphere MDM is still pointing to the temporary database of the new release.

**Note:** As of version 11, customizations are deployed as composite bundle archives (CBAs) with no disruption to the deployed application itself.

**Step 3:** Create and configure any customized JMS resources, such as new notification queues or topics, and their bindings, if they exist.

**Step 4:** Execute the regression test suite to test all of the implemented functionality of the solution. Exercise all clients invoking the capabilities and ensure coverage for all of your customizations.

**Step 5:** Back up the existing MDM database. For example, back up the existing version 8.0.1 database when upgrading InfoSphere MDM from version 8.0.1 to 9.0.2.

**Step 6:** Upgrade the existing database to the level of the new release by applying InfoSphere MDM database upgrade scripts. If the upgrade involves multiple releases, the database upgrade may need to be applied incrementally, starting with the earliest version and ending with the latest version.

**Note:** You may need to first upgrade your existing MDM database infrastructure to the new database software level, if a different version is required for new MDM product version.

**Step 7:** Repopulate the content of the APPSOFTWARE, APPDEPLOYMENT, and CONFIGELEMENT tables for the upgraded InfoSphere MDM database. This can be done by exporting data from the temporary InfoSphere MDM database of the new release and then importing it into the upgraded InfoSphere MDM database. At this moment, the upgraded InfoSphere MDM database should have the same configuration as the temporary database. Pay attention to records in APPSOFTWARE, APPDEPLOYMENT, and APPINSTANCE to make sure that they are consistent with the running InfoSphere MDM instance. If the application runs on a cluster, make sure the associated entries in CONFIGELEMENT are configured properly to support the InstanceIdPK uniqueness per cluster member of the application instance.

**Note:** This step is not necessary if upgrading from version 10.1 to version 11 as the upgrade scripts already populate these APPSOFTWARE, APPDEPLOYMENT, and CONFIGELEMENT tables with a deployment id =1000.

**Step 8:** Change the data source of the newly installed InfoSphere MDM to point to the upgraded database. Since the upgraded database has the same configuration as the temporary database and the InfoSphere MDM application has been tested thoroughly with the temporary database, all issues related to database configurations should already be resolved at this moment.

**Step 9:** Execute the regression test suite once again to test all implemented functionality of the solution. Exercise all clients (see step 10) invoking the capabilities and ensure coverage for all of your customizations.

**Step 10:** Update all existing client applications to use the newly installed application, and execute tests from client applications (IT and UAT only). These include all web services clients, JMS clients, RMI clients, etc.
On Testing
A comprehensive test effort is critical to completing the upgrade process and successfully going live on a new product release. As such, it is important that you consider the testing element of the upgrade as a major task. Typically, functional regression testing is performed in the DEV environment; integration testing and performance testing is performed in the IT environment; and user acceptance testing is performed in the UAT environment. You may want to leverage automated testing tools to augment with human testing efforts to help the effort to go more quickly.

To prepare test scripts for use during the testing cycles, you should begin with the test scripts used during the original implementation. Augment these scripts to include testing of the new features and functionality. If these materials do not exist from the original implementation, you should create them and store them in a document repository. This material can save significant time in preparation for testing future upgrades.

The Server Migration Pass is a recursive process until an acceptable result is reached. If any problems are discovered and fixed in any step, a complete retest will be necessary.

Best Practices

• Test the server migration process thoroughly before moving to the production environment in order to minimize time needed for the production upgrade.

Final Move to Production

With acceptable test results from the Server Migration Pass, we are ready for the Final Move to Production stage. To achieve near-zero production down time, a high availability (HA) infrastructure, including a high availability application server and database server, must exist. If it does not exist, you must create one. In the example presented below, we will assume that the HA infrastructure for the application server and database server already exists in the production environment.

Note: High availability of the application server is achieved by using two instances of InfoSphere MDM. High availability of the database server is achieved by using two database servers, one acting as primary and the other acting as secondary in an active-standby mode.

Prepare Production Servers for Upgrade
During the server migration pass, you tested the new release of the application and all of your customizations with a temporary database so that you could address any issues at that time, before switching to the production database. Now, the following activities should be performed to prepare your production servers for upgrade. These steps should be familiar to you now:

**Step 1:** Execute the Base Installation Pass for the Production server in the new Production environment.

**Step 2:** Run all InfoSphere MDM customization database scripts on the temporary database that was created in the Base Installation Pass.

**Step 3:** Deploy your customizations onto the production server. Recall that for versions prior to version 11 this will involve deploying the InfoSphere MDM EAR with all customizations incorporated into it. As of version 11, this will involve the deployment of one or more CBAs. At this point, the new release of InfoSphere MDM is still pointing to the temporary database of the new release in the new production environment.

**Step 4:** Create and configure any customized JMS resources, such as new notification queues or topics, and their bindings, if they exist.

**Step 5:** Execute the regression test suite once again to test all implemented functionality of the solution. Include tests that exercise SOA interfaces used by your solution (RMI or web services), and custom web UI applications.

**Execute Production Servers Upgrade**

After the completion of the Code Migration Pass, the Server Migration Pass, and the production server preparation, we are finally ready for the final move – upgrading production servers. Before the upgrade takes place on production servers, the new release of InfoSphere MDM that is installed on the production server should have been connected to a temporary database and it should have been fully tested with the temporary database. We assume that the high availability solution of application servers has been implemented with multiple InfoSphere MDM instances, and the high availability solution for the database has been implemented with two databases.

Data synchronization between these two databases can be achieved through HA tools available for the given type of database server. For example, you may use the DB2 high availability tool for DB2 databases. For details of these database synchronization options, you may refer to references listed in the reference section of this article.

**Stepping Through an Example**

We will next describe the process to be performed on the production server using an example of the upgrade from InfoSphere MDM 8.0.1 to InfoSphere MDM 9.0.2 with two InfoSphere MDM application instances in a cluster.

**Starting Environment:** At the beginning of the production server upgrade, there are two instances of InfoSphere MDM 8.0.1 running in separate environments, and two MDM 8.0.1 databases residing on different servers — one acting as primary and the other secondary. Real-time synchronization occurs between these two databases. The two newly installed InfoSphere
MDM 9.0.2 instances are connected to two temporary MDM 9.0.2 databases, as shown in Figure 2.

*Figure 2: Production environment starting point*
Step 1: Stop the data synchronization process running between the two MDM 8.0.1 databases, as shown in Figure 3. This will allow one of the MDM 8.0.1 databases to be upgraded to MDM 9.0.2 while the other is still serving MDM 8.0.1 applications. The standby MDM 8.0.1 database will be upgraded to InfoSphere MDM 9.0.2 first.
Step 2: Back up the standby MDM 8.0.1 database and then upgrade it to MDM 9.0.2, as shown in Figure 4. This will prepare the second MDM 8.0.1 database for InfoSphere MDM 9.0.2 applications. The existing InfoSphere MDM 8.0.1 applications will continue to point to the first MDM 8.0.1 database during the upgrade of the second MDM 8.0.1 database.

The database upgrade activities include running all database upgrade scripts from MDM 8.0.1 to MDM 9.0.2, deleting old contents in tables APPSOFTWARE, APPINSTANCE, APPDEPLOYMENT, and CONFIGELEMENT, and then repopulating them with contents for InfoSphere MDM 9.0.2. These four tables can be populated by exporting data from the temporary MDM 9.0.2 database and then importing it into the upgraded database or by running the Management Agent. **Note:** This is not necessary if upgrading from version 10.1 to version 11 as the upgrade scripts already populate these APPSOFTWARE, APPDEPLOYMENT, and CONFIGELEMENT tables with a deployment id =1000.

If any data changes in the APPINSTANCE and CONFIGELEMENT table are needed, it is recommended that these changes be done at this stage in order to properly support the primary key uniqueness required amongst multiple InfoSphere MDM application instances on a cluster.

![Figure 4: Upgrade the second MDM 8.0.1 database to MDM 9.0.2](image-url)
Step 3: In this step, switch the data source for the InfoSphere MDM 9.0.2 servers from the temporary database to the upgraded MDM 9.0.2 database. At this point, both instances are pointing to the upgraded MDM 9.0.2 database. The existing MDM 8.0.1 applications will continue to point to the primary MDM 8.0.1 database during the upgrade process of the secondary MDM 8.0.1 database, as shown in Figure 5. Now, testing should be done to ensure InfoSphere MDM 9.0.2 applications work correctly functionally with the upgraded MDM 9.0.2 database.

Figure 5: Switch database for InfoSphere MDM 9.0.2 instances
Step 4: Run data synchronization between the MDM 8.0.1 database and the upgraded MDM 9.0.2 database, as shown in Figure 6. This will allow the upgraded MDM 9.0.2 database to catch up any data changes made during the upgrade of the standby MDM 8.0.1 database to MDM 9.0.2.

Figure 6: Run data synchronization between the MDM 9.0.2 and MDM 8.0.1 databases
Step 5: Turn off data synchronization between the remaining MDM 8.0.1 database and the upgraded MDM 9.0.2 database and then switch over the traffic from the client applications from InfoSphere MDM 8.0.1 to InfoSphere MDM 9.0.2, as shown in Figure 7. At this point, all client applications that used to access InfoSphere MDM 8.0.1 applications will be connected to InfoSphere MDM 9.0.2 applications. Both InfoSphere MDM 9.0.2 instances are pointing to the MDM 9.0.2 database. Both MDM 8.0.1 servers will be stopped.

Figure 7: Switch client traffic from InfoSphere MDM 8.0.1 to InfoSphere MDM 9.0.2
Step 6: Upgrade the other MDM 8.0.1 database to MDM 9.0.2, as shown in Figure 8. At this point, both InfoSphere MDM 9.0.2 instances still point to the first updated MDM 9.0.2 database.

Figure 8: Upgrade the second InfoSphere MDM 8.0.1 database to InfoSphere MDM 9.0.2
Step 7: Turn on data synchronization between the two upgraded MDM 9.0.2 databases, as shown in Figure 9. This will allow the second upgraded MDM 9.0.2 database to catch up any data changes that occurred after switching client traffic from the version 8.0.1 server to the version 9.0.2 server.

Figure 9: Run data synchronization between the two upgraded InfoSphere MDM 9.0.2 databases
Step 8: Switch both MDM 9.0.2 instances to point to the primary MDM 9.0.2 database, as shown in Figure 10. The data synchronization process from the primary to the secondary MDM 9.0.2 database will be resumed.

Figure 10: Switch MDM 9.0.2 instances to point to the primary MDM 9.0.2 database
Step 9: Retire all MDM 8.0.1 instances. The target topology for the production environment has now been reached, as shown in Figure 11.

![Diagram of Production Environment Post-Upgrade]

**Figure 11: Production environment post-upgrade**

### Best Practices

- Leverage high availability infrastructure to achieve near zero production downtime.

- Using redundant application servers and databases, one of the existing MDM Servers and databases will remain to serve production traffic while the other existing database is being upgraded.
Summary

Achieving an objective of near-zero production downtime during an upgrade of InfoSphere MDM requires careful planning and execution. Throughout this document, the activities in each of these passes have been described and illustrated with examples.

Best Practices

- Test your new installation thoroughly to verify all required components are working as expected.

- Keep current. Apply all available product fix packs once the base installation has been completed.

- Migrate the code to your existing customizations and test thoroughly in the development environment to isolate code migration issues from application and database upgrade issues that may be uncovered as part of the server migration activity.

- Review your code and take the appropriate actions to resolve and remove the use of deprecated APIs (ex. utility classes) you may have introduced in your customizations.

- Add new customizations as a separate phase of the project, after your existing ones have been tested and work as designed.

- Test the server migration process thoroughly applying it to production servers to minimize time needed for production upgrade.

- Leverage high availability infrastructure to achieve near zero production downtime.

- Using redundant application servers and databases, one of the existing MDM Servers and databases will remain to serve production traffic while the other existing database is being
Further reading

- InfoSphere Master Data Management version 10.0 Information Center
- InfoSphere Master Data Management version 10.1 Information Center
- InfoSphere Master Data Management version 11 Information Center
- Monitoring and tuning InfoSphere Master Data Management Server: Tune the MDM Server environment for high performance access to master data
- Achieving high availability and scalability with InfoSphere MDM Server
- High Availability and Disaster Recovery Options for DB2 for Linux, UNIX, and Windows

Contributors

Stephanie Hazlewood
Senior Product Architect
InfoSphere MDM
IBM Software Group

Dennis Shi
Managing Consultant
InfoSphere MDM Lab Services
IBM Software Group

Bill Xu
Senior Technical Specialist
InfoSphere MDM Lab Services
IBM Software Group
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 does 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.

COPYRIGHT LICENSE: © Copyright IBM Corporation 2012. All Rights Reserved.

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.