IBM Interact
Version 9 Release 1
May 8, 2014

Fix Pack Guide

Note
Before using this information and the product it supports, read the information in "Notices" on page 19.

This edition applies to version 9, release 1, modification 0 of IBM Interact and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Overview of the IBM Interact Fix Pack 2

The IBM® Interact 9.1.0 Fix Pack 2 provides a number of improvements over the base 9.1.0 installation, including fixed defects and improved results when you process event pattern data. This guide describes the features added to Interact by this fix pack.

Stand-alone event pattern ETL process.

In Interact, all event pattern data for a given AudienceID is stored as a single collection in the runtime database tables. To perform any SQL queries or reporting based on event patterns, this new ETL process is required to break up the collection of event pattern data into tables in a target database. To accomplish this, the stand-alone ETL process takes event pattern data from the Interact runtime database tables, processes it on the schedule you specify, and stores it in the target database where it is available for SQL queries or additional reporting.

System schema changes to accommodate the ETL process

The stand-alone event pattern ETL process stores the processed event pattern data in a target database for further use in SQL queries or reporting. The target database must be updated with four new tables that are required to store the data. These tables are documented here for reference in retrieving the data in your own reporting process.

Offer deduplication across offer attributes

Using the Interact application programming interface (API), two API calls deliver offers: `getOffers` and `getOffersForMultipleInteractionPoints`. `getOffersForMultipleInteractionPoints` can prevent the return of duplicate offers at the *OfferID* level, but cannot deduplicate offers across offer category. So, for example, for Interact to return only one offer from each offer category, a workaround was previously required. With the introduction of two parameters to the `startSession` API call, offer deduplication across offer attributes, such as category, is now possible.

Chapter 2. Event pattern ETL process

To process large amounts of IBM Interact event pattern data and to make that data available for queries and reporting purposes, you can install a stand-alone Extract, Transform, Load (ETL) process on any supported server for optimal performance.

In Interact, all event pattern data for a given AudienceID is stored as a single collection in the runtime database tables. The AudienceID and pattern state information is stored as a Binary Large Object (BLOB). To perform any SQL queries or reporting based on event patterns, this new ETL process is necessary to break up the object into tables into a target database. To accomplish this, the stand-alone ETL process takes event pattern data from the Interact runtime database tables, processes it on the schedule you specify, and stores it in the target database where it is available for SQL queries or additional reporting.

In addition to moving and transforming event pattern data to the target database, the stand-alone ETL process also synchronizes the data in the target database with the most current information in your Interact runtime database. For example, if you delete an event pattern in the Interact runtime, that event pattern's processed data is removed from the target database the next time the ETL process runs. Event pattern state information is kept up to date as well. So the information stored about event patterns in the target database is solely current data, not historical information.

Installing the stand-alone ETL process

To process large numbers of event pattern ETL processes, you can install the ETL process on a standalone server for optimal performance. You install the stand-alone ETL process using the regular IBM EMM installer.

Before you can install the stand-alone Interact Event Pattern ETL process, you must have completed the following tasks:

- Install a complete IBM Interact setup, including the IBM Marketing Platform server and one or more Interact run-time servers.

  The installation process is described in detail in the *Interact Installation Guide*.
- Install and configure the data source in which you want the event pattern ETL process to store its data. This may be the same data source in which the Interact runtime tables are stored, or it may be a different data source for performance reasons.
- Collect and have available the Marketing Platform server networking information, as described in the *Interact Installation Guide*. You need this information during this installation process.
- Be sure that a supported Java run-time environment is installed on the server where you are installing the ETL process.
- Connect to the server on which you are installing the ETL process with administrator or root privileges.

When you complete this task, the files necessary to run the stand-alone ETL process will be available on the server. You will still need to configure the process before you can run it.

1. On the server where you want to run the stand-alone event pattern ETL process, copy the IBM EMM master installation program for the operating system along with the IBM EMM Interact installer. Remember that both the master installer and the Interact installer must be in the same directory, and that you must run the installer as a user with administrator-level privileges on the server.
2. Follow the instructions in the *IBM Interact Installation Guide* to launch the master installation program. Be sure to provide the connection information for the IBM Marketing Platform server used by your runtime servers and design time server.
3. When the IBM Interact installer launches and displays the Interact Components page, select only the **Interact Event Pattern ETL** option to install.
4. Follow the prompts until you complete the installation.
5. On the server where you installed the ETL process, locate the `<Interact_Home>/PatternStateETL/ddl` directory.
6. Using your database management software, run the appropriate script in the `DDL` directory against the database that you will use as the target database for the output from the ETL process.

   The scripts in this directory create four tables in the target database that are required to use the ETL process. Depending on the target database you are using, run one of the following scripts:

   - `aci_evpattab_db2.ddl` if the target database is IBM DB2.
   - `aci_evpattab_ora.ddl` if the target database is Oracle.
   - `aci_evpattab_sqlsvr.ddl` if the target database is Microsoft SQL Server.

You have now installed the event pattern ETL process on the server. If you accepted the default installation directory during installation, you can find the installed files in `C:\IBM\EMM\Interact` on a supported Microsoft Windows platform, or in `/IBM/EMM/Interact` on a supported UNIX-like operating system.

To continue with the stand-alone event pattern ETL process, you need to configure the process by modifying files on the ETL process server, and in the Marketing Platform configuration pages.

Configuring the stand-alone ETL process

After you install the Interact stand-alone ETL process, you need to configure the process by modifying files on the ETL process server, and in the Marketing Platform configuration pages.

To configure the ETL process, there are files in the Interact home directory on the ETL process server, to indicate where the necessary Java runtime files are found, as well as other environment variables. Then you need to connect to the IBM Marketing Platform server associated with this installation and use the configuration pages there to set up the necessary properties to run the ETL process.

1. On the server on which you have installed the stand-alone ETL process, open the following file in any text editor: `<Interact_home>\PatternStateETL\bin\setenv.bat` on Microsoft Windows, or `<Interact_home>\PatternStateETL\bin\setenv.sh` on a UNIX-like operating system.
   a. Complete the line that reads `set JAVA_HOME=[CHANGE ME]`, changing `[CHANGE ME]` to the actual path to the 64-bit Java runtime you want to use.

Note: Although the IBM EMM installer provides a Java runtime in `<Interact_home>\..\jre`, such as `C:\IBM\EMM\jre`, this is a 32-bit Java runtime used only for installation. This runtime is not suitable for running the ETL process. Install a supported 64-bit Java runtime if one is not already installed, and update the `setenv` file to use that runtime.

b. Complete the line that reads `set JDBCDRIVER_CP=` with the actual location of the JDBC driver for the connection to the database containing the system tables. For example, if you were connecting to an Oracle database, you might specify the path to a local copy of `ojdbc6.jar`.

2. In a supported web browser, connect to the IBM Marketing Platform server associated with this installation and log in using administer-level credentials.
3. Open the Configuration page by clicking **Settings > Configuration** in the toolbar.

   The Configuration page shows the Configuration Categories tree.
4. Navigate to **interact | ETL** in the Configuration Categories tree.
5. Click **PatternStateETLConfig** in the tree to create a new Pattern State ETL configuration.

   In the right pane, complete the following information:

   - **New category name**. Provide a name that uniquely identifies this configuration. Note that you must provide this exact name when you run the stand-alone ETL process. For convenience in specifying this name on the command line, you may want to avoid a name containing spaces or punctuation, such as `ETLProfile1`.
   - **runOnceADay**. Determines whether the stand-alone ETL process in this configuration should run once each day. Valid answers are **Yes** or **No**. If you answer **No** here, the **processSleepIntervalInMinutes** determines the run schedule for the process.
   - **preferredStartTime**. The preferred time at which the stand-alone ETL process should start. Specify the time in the format HH:MM:SS AM/PM, as in `01:00:00 AM`.
   - **preferredEndTime**. The preferred time at which the stand-alone ETL process should stop. Specify the time in the format HH:MM:SS AM/PM, as in `08:00:00 AM`.
   - **processSleepIntervalInMinutes**. If you have not configured the stand-alone ETL process to run once a day (as specified in the **runOnceADay** property), this property specifies the interval between ETL process runs. For example, if you specify 15 here, the stand-alone ETL process will wait for 15 minutes after it stops running before starting the process again.
   - **maxJDBCInsertBatchSize**. The maximum number of records of a JDBC batch before committing the query. By default, this is set to 5000. Note that this is not the maximum number of records that the ETL processes in one iteration. During each iteration, the ETL processes all available records from the UACI_EVENTPATTERNSTATE table. However, all those records are broken into **maxJDBCInsertSize** chunks.
   - **maxJDBCFetchBatchSize**. The maximum number of records of a JDBC batch to fetch from the staging database.

     You may need to increase this value to tune the performance of the ETL.
   - **communicationPort**. The network port on which the standalone ETL process listens for a stop request. Under normal circumstances, there should be no reason to change this from the default value.

- **queueLength**. A value used for performance tuning. Collections of pattern state data are fetched and transformed into objects that are added to a queue to be processed and written to the database. This property controls the size of the queue.
- **completionNotificationScript**. Specifies the absolute path to a script to run when the ETL process is completed. If you specify a script, three arguments are passed to the completion notification script: start time, end time, and total number of event pattern records processed. The start time and end time are numeric values representing number of milliseconds elapsed since 1970.

6. When you have finished completing the configuration, click **Save**. When you save the configuration, two additional categories are automatically created in the tree underneath the new configuration: RuntimeDS, and TargetDS. Use these categories to specify the data source where the stand-alone ETL process should retrieve the data it will process (the database containing the Interact runtime tables), and the data source where the results will be stored.
7. Configure the **Interact | ETL | patternStateETL | <patternStateETLName> | RuntimeDS** and **Interact | ETL | patternStateETL | <patternStateETLName> | TargetDS** categories for the ETL configuration.

   The two categories determine the data sources for retrieving and storing the event pattern data used by the ETL process.

   Note: The data source that you specify for the **TargetDS** configuration may be the same data source in which the Interact runtime tables are stored, or it may be a different data source for performance reasons.

   a. Click the category (**RuntimeDS** or **TargetDS**) that you want to configure.
   b. In the right pane, click **Edit Settings** and complete the following fields:
      - **type**. A list of the supported database types for the data source you are defining.
      - **dsname**. The JNDI name of the data source. This name must also be used in the user's data source configuration to ensure that the user has access to the target and runtime data sources.
      - **driver**. The name of the JDBC driver to us, such as any of the following:

        Oracle: `oracle.jdbc.OracleDriver`

        Microsoft SQL Server: `com.microsoft.sqlserver.jdbc.SQLServerDriver`

        IBM DB2: `com.ibm.db2.jcc.DB2Driver`
      - **serverUrl**. The data source URL, such as any of the following:

        Oracle: `jdbc:oracle:thin:@ <your_db_host>:<your_db_port>:<your_db_service_name>`

        Microsoft SQL Server: `jdbc:sqlserver:// <your_db_host>:<your_db_port> ;databaseName= <your_db_name>`

        IBM DB2: `jdbc:db2:// <your_db_host>:<your_db_port>/<your_db_name>`
      - **connectionpoolSize**. A value indicating the size of the connection pool, provided for performance tuning. Pattern state data is read and transformed concurrently depending upon the available database connections. Increasing the connection pool size allows for more concurrent database connections, subject to limitations of memory and database read/write capabilities. For example, if this value is set to 4, four jobs will run concurrently. If you have a large amount of data, you might need to increase this value to a number such as 10 or 20, as long as sufficient memory and database performance is available.

- **schema**. The name of the database schema to which this configuration is connecting.
- **connectionRetryPeriod**. The `ConnectionRetryPeriod` property specifies the amount of time in seconds Interact automatically retries the database connection request on failure. Interact automatically tries to reconnect to the database for this length of time before reporting a database error or failure. If the value is set to 0, Interact retries indefinitely; if the value is set to -1, no retry is attempted.
- **connectionRetryDelay**. The `ConnectionRetryDelay` property specifies the amount of time in seconds Interact waits before it tries to reconnect to the database after a failure. If the value is set to -1, no retry is attempted.

Save the changes when you are done specifying both the runtime and target data sources.

8. Still in IBM Marketing Platform server, click **Settings > Users** in the toolbar.
9. Edit the user that will be running the stand-alone ETL process and click **Edit Data Sources**.
10. Define data sources for the user to match the **TargetDS** and **RuntimeDS** categories you just defined for the ETL category. The Data Source name you specify for the user data source must match the value of the dsname property for TargetDS or RuntimeDS configuration. The event pattern state ETL read the user name and password you specify here to connect to the database during processing.

You have now configured the Marketing Platform for use with the event pattern ETL process. Be aware that any changes you make to the ETL configuration other than the communication port are automatically implemented in the next run of the ETL process. There is no need to restart the ETL process after you change the configuration, unless you specify a new communication port.

With the installation and configuration of the event pattern ETL process completed, you are now ready to run the process.

Running the stand-alone ETL process

When you launch the stand-alone ETL process on a server, it runs continuously in the background until stopped. The process follows the instructions in the Marketing Platform configuration properties to determine frequency, database connections, and other details during its operation.

Before you can run the stand-alone ETL process, you must have installed the process on a server, and configured both the files on the server and in the Marketing Platform correctly for your configuration.

Note:

If you are running the ETL process on Microsoft Windows for a language other than US English, use `chcp` at the command prompt to set the code page for the language you are using. For example, you might use any of the following codes: ja_jp=932, zh_cn=936, ko_kr=949, ru_ru=1251 and for de_de, fr_fr, it_it, es_es, pt_br, use 1252. To ensure proper character display, use the `chcp` command in the Windows command prompt prior to launching the ETL process.

After you have installed and configured the stand-alone ETL process, you are ready to launch the process.

1. Open a command prompt on the server where the ETL process is installed.
2. Navigate to the `<Interact_home>/PatternStateETL/bin` directory that contains the executable files for the ETL process.
3. Run the `command.bat` file (on Microsoft Windows) or `command.sh` file (on UNIX-like operating systems) with the following parameters:
   - `-u <username>`. This value must be a valid Marketing Platform user, and you must have configured that user with access to the **TargetDS** and **RuntimeDS** datasources that the ETL process will use.
   - `-p <password>`. Replace *<password>* with the password matching the user you specified. If the password for this user is blank, specify two double quotes (as in `-p ""`). The password is optional when you run the command file; if you omit the password with the command, you are prompted to enter it when the command runs.
   - `-c <profileName>`. Replace *<profileName>* with the exact name you specified in the Marketing Platform in the **Interact | PatternStateETL** configuration you created.

     The name you enter here must match the value you specified in the **New category name** field when you created the configuration.
   - `start`. The start command is required to start the process.

   The complete command to start the process would therefore take the following form:

   ```
   command.bat -u <username> -p <password> -c <profileName> start
   ```

The stand-alone ETL process runs, and continues to run in the background until you stop the process or until the server is restarted.

Note:

The first time you run the process, the accumulated event pattern data may take a considerable amount of time to run. Subsequent times that the process runs will work with only the most recent set of event pattern data and will take less time to complete.

Be aware that you can also provide the `help` argument to the `command.bat` or `command.sh` file to see all available options, as in the following example:

```
command.bat help
```

Stopping the stand-alone ETL process

When you launch the stand-alone ETL process on a server, it runs continuously in the background until stopped.

1. Open a command prompt on the server where the ETL process is installed.
2. Navigate to the `<Interact_home>/PatternStateETL/bin` directory that contains the executable files for the ETL process.
3. Run the `command.bat` file (on Microsoft Windows) or `command.sh` file (on UNIX-like operating systems) with the following parameters:
   - `-u <username>`. This value must be a valid Marketing Platform user, and you must have configured that user with access to the **TargetDS** and **RuntimeDS** data sources that the ETL process will use.

- `-p <password>`. Replace *<password>* with the password matching the user you specified. If the password for this user is blank, specify two double quotes (as in `-p ""`). The password is optional when you run the command file; if you omit the password with the command, you are prompted to enter it when the command runs.
- `-c <profileName>`. Replace *<profileName>* with the exact name you specified in the Marketing Platform in the **Interact | PatternStateETL** configuration you created.

  The name you enter here must match the value you specified in the **New category name** field when you created the configuration.
- `stop`. The stop command is required to stop the process. If you use this command, any ongoing ETL operation will complete before the process shuts down.

  To shut down the ETL process without waiting for any ongoing operations to complete, use `forcestop` instead of `stop`.

The complete command to start the process would therefore take the following form:

```
command.bat -u <username> -p <password> -c <profileName> stop
```

The stand-alone ETL process stops.

Note that you can also provide the `help` argument to the `command.bat` or `command.sh` file to see all available options, as in the following example:

```
command.bat help
```

Chapter 3. System schema changes in this fixpack

The event pattern ETL process provided by this fixpack creates a number of tables in the Interact ETL target database, which is the repository for the results of the event pattern ETL processing to make that data available for reporting.

When you install the Event Pattern ETL process that is provided as part of this Interact fix pack, a set of SQL scripts were also installed in the following location:

```
<Interact_home>\PatternStateETL\ddl
```

When you run one of the scripts provided (`aci_evpattab_db2.sql`, `aci_evpattab_ora.sql`, `aci_evpattab_sqlsvr.sql`) against the target data source for the ETL process , you create four tables into which the output from the stand-alone event pattern ETL process is stored. Each of these tables and their schemas are described here.

System schema reference

When you install the standalone event pattern ETL process, you must run the provided SQL script to create four tables in the target database (the data source identified as TargetDS in the IBM Marketing Platform configuration settings). The processed event pattern data is stored in these tables, and you can retrieve the information using the descriptions here for reporting purposes.

UACI_ETLPatternEventInfo

This table contains the most recent event information for each specified event pattern, such as whether the pattern is enabled, the start and end time, and so on.

Field	Type	Length	Null?	Description
PatternEventId	INT64		false	The ID of this event pattern.
Name	VARCHAR	64	false	The name of the event pattern. Can be any text characters, minus standard disallowed special name characters.
CategoryId	INT64		false	Unique ID of the category that contains the event. Must be a globally unique positive integer within the internalIDLowerLimit and internalIDUpperLimit configuration parameter values for generated values.
CategoryName	VARCHAR	64	false	Name of the category. Can be any text characters, minus standard disallowed special name characters.

Field	Type	Length	Null?	Description
ICId	INT64		false	The unique ID of the Interact Channel to which the event pattern belongs. Must be a globally unique positive integer within the internalIDLowerLimit and internalIDUpperLimit configuration parameter values for generated values.
ICName	VARCHAR	64	false	The name of the Interact Channel to which the event pattern belongs.
SourceDB	VARCHAR	128	true	The data source from which this event pattern was transferred by the ETL process. This is usually the Interact runtime data source as specified in the ETL configuration file.
Type	INT32		true	The type of event pattern
UpdateTime	INT64		false	The date and time in the format of yyyyMMddhhmmss that this event pattern record was last updated.
StartTime	INT64		true	The date and time in the format of yyyyMMddhhmmss at which this event pattern starts to be in effect.
EndTime	INT64		true	The date and time in the format of yyyyMMddhhmmss at which this event pattern stops being in effect.
resetDuration	INT64		true	The extended time span during which the pattern retains its true state before it resets and begins evaluating events again.
isEnabled	INT32		true	Boolean value indicating whether the event pattern is active or not.

UACI_ETLPatternStateItem

This table contains pattern state instance details.

Field	Type	Length	Null?	Description
PatternStateInstanceId	INT64		false	Unique ID of the event pattern state instance. Must be a globally unique positive integer within the internalIDLowerLimit and internalIDUpperLimit configuration parameter values for generated values.
EventId	INT64		false	Unique ID of the event. Must be a globally unique positive integer within the internalIDLowerLimit and internalIDUpperLimit configuration parameter values for generated values.
EventCount	INT32		false	

Field	Type	Length	Null?	Description
EventWeight	INT64		true	

UACI_ETLPatternStateRun

This table tracks the details of each ETL run.

Field	Type	Length	Null?	Description
Starttime	DATETIME		false	The date and time in the format of yyyyMMddhhmmss that this event pattern state run began.
Endtime	DATETIME		true	The date and time in the format of yyyyMMddhhmmss that this event pattern state run was completed.
Status	INT32		false	Indicates the current status of the given ETL operation, with values matching one of the following: • 0: Preparing • 1: In Progress • 2: Completed with failures • 3: Completed Successfully

UACI_ETLPatternState

Holds pattern states instances for a given AudienceID.

Field	Type	Length	Null?	Description
PatternStateInstanceId	INT64		false	
AudienceId	VARCHAR	128	false	
AudienceLevel	VARCHAR	64	false	
PatternId	INT64		false	
State	INT32		false	The current state of the event pattern, from one of the following values: • 1: pattern triggered • 0: not triggered • -1: expired • -2: disabled
TotalCountScore	INT64		false	
ExpectedCountScore	INT64		false	
UpdateTime	INT64		false	
ActivationTime	INT64		true	

Chapter 4. Offer deduplication across offer attributes

Using the Interact application programming interface (API), two API calls deliver offers: `getOffers` and `getOffersForMultipleInteractionPoints`. `getOffersForMultipleInteractionPoints` can prevent the return of duplicate offers at the *OfferID* level, but cannot deduplicate offers across offer category. So, for example, for Interact to return only one offer from each offer category, a workaround was previously required. With the introduction of two parameters to the `startSession` API call, offer deduplication across offer attributes, such as category, is now possible.

This list summarizes the parameters that were added to the `startSession` API call. For more information about these parameters or any aspect of the Interact API, see the *IBM Interact Administrator's Guide*, or the Javadoc files included with your Interact installation in `<Interact_Home>/docs/apiJavaDoc`.

- `UACIOfferDedupeAttribute`. To create a `startSession` API call with offer deduplication, so that the subsequent `getOffer` calls return only one offer from each category, you must include the `UACIOfferDedupeAttribute` parameter as part of the API call. You can specify a parameter in the `name,value,type` format, as shown here:

  `UACIOfferDedupeAttribute,<attributeName>,string`

  In this example, you would replace *<attributeName>* with the name of the offer attribute you want to use as the criterion for deduplication, such as Category.

  Note: Interact examines the offers that have the same attribute value you specify (such as Category) and deduplicate to remove all but the offer that has the highest score. If the offers that have the duplicate attribute also have identical scores, Interact returns a random selection from among the matching offers.
- `UACINoAttributeDedupeIfFewerOf`. When you include the `UACIOfferDedupeAttribute` in the `startSession` call, you can also set this `UACINoAttributeDedupeIfFewerOf` parameter to specify the behavior in cases where the offer list after deduplication no longer contains enough offers to satisfy the original request.

  For example, if you set `UACIOfferDedupeAttribute` to use the offer category to deduplicate offers, and your subsequent getOffers call requests that eight offers be returned, deduplication might result in fewer than eight eligible offers. In that case, setting `UACINoAttributeDedupeIfFewerOf` parameter to true would result in adding some of the duplicated to the eligible list to satisfy the requested number of offers. In this example, if you set the parameter to false, the number of offers that are returned would be fewer than the requested number.

  `UACINoAttributeDedupeIfFewerOf` is set to true by default.

For example, suppose you specified as a `startSession` parameter that the deduplication criterion is offer Category, as shown here:

```
UACIOfferDedupeAttribute, Category,
string;UACINoAttributeDedupeIfFewerOffer, 0, string
```

These parameters together cause Interact to deduplicate offers based on the offer attribute "Category," and to return only the deduplicated offers even if the

resulting number of offers is fewer than requested (`UACINoAttributeDedupeIfFewerOffer` is false).

When you issue a `getOffers` API call, the original set of eligible offers might include these offers:

- Category=Electronics: Offer A1 with a score of 100 and Offer A2 with a score of 50.
- Category=Smartphones: Offer B1 with a score of 100, Offer B2 with a score of 80, and offer B3 with a score of 50.
- Category=MP3Players: Offer C1 with a score of 100, Offer C2 with a score of 50.

In this case, there were two duplicate offers that match the first category, three duplicate offers that match the second category, and two duplicate offers that match the third category. The offers that are returned would be the highest scoring offers from each category, which are Offer A1, Offer B1, and Offer C1.

If the `getOffers` API call requested six offers, this example set `UACINoAttributeDedupeIfFewerOffer` to false, so only three offers would be returned.

If the `getOffers` API call requested six offers, and this example omitted the `UACINoAttributeDedupeIfFewerOffer` parameter, or specifically set it to true, some of the duplicate offers would be included in the result to satisfy the requested number.

Before you contact IBM technical support

If you encounter a problem that you cannot resolve by consulting the documentation, your company's designated support contact can log a call with IBM technical support. Use these guidelines to ensure that your problem is resolved efficiently and successfully.

If you are not a designated support contact at your company, contact your IBM administrator for information.

Information to gather

Before you contact IBM technical support, gather the following information:

- A brief description of the nature of your issue.
- Detailed error messages that you see when the issue occurs.
- Detailed steps to reproduce the issue.
- Related log files, session files, configuration files, and data files.
- Information about your product and system environment, which you can obtain as described in "System information."

System information

When you call IBM technical support, you might be asked to provide information about your environment.

If your problem does not prevent you from logging in, much of this information is available on the About page, which provides information about your installed IBM applications.

You can access the About page by selecting **Help > About**. If the About page is not accessible, check for a `version.txt` file that is located under the installation directory for your application.

Contact information for IBM technical support

For ways to contact IBM technical support, see the IBM Product Technical Support website: (http://www.ibm.com/support/entry/portal/open_service_request).

Note: To enter a support request, you must log in with an IBM account. This account must be linked to your IBM customer number. To learn more about associating your account with your IBM customer number, see **Support Resources > Entitled Software Support** on the Support Portal.

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 about the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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
170 Tracer Lane
Waltham, MA 02451
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 document 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 contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not

been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

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

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Privacy Policy and Terms of Use Considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. A cookie is a piece of data that a web site can send to your browser, which may then be stored on your computer as a tag that identifies your computer. In many cases, no personal information is collected by these cookies. If a Software Offering you are using enables you to collect personal information through cookies and similar technologies, we inform you about the specifics below.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's user name, and other personal information for purposes of session management, enhanced user usability, or other usage tracking or functional purposes. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

Various jurisdictions regulate the collection of personal information through cookies and similar technologies. If the configurations deployed for this Software Offering provide you as customer the ability to collect personal information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for providing notice and consent where appropriate.

IBM requires that Clients (1) provide a clear and conspicuous link to Customer's website terms of use (e.g. privacy policy) which includes a link to IBM's and Client's data collection and use practices, (2) notify that cookies and clear gifs/web beacons are being placed on the visitor's computer by IBM on the Client's behalf along with an explanation of the purpose of such technology, and (3) to the extent required by law, obtain consent from website visitors prior to the placement of cookies and clear gifs/web beacons placed by Client or IBM on Client's behalf on website visitor's devices

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Online Privacy Statement at: http://www.ibm.com/privacy/details/us/en section entitled "Cookies, Web Beacons and Other Technologies."

Printed in USA