

Express Hints

Version 3.1

User's Guide



HLS Technologies, Inc.

3322 Sturbridge Lane

Sugar Land, Texas 77479-2223

Phone (281) 265-3004

Toll-free (888) 494-9019

Fax (281) 265-3006

sales@hlstechnologies.com

<http://www.hlstechnologies.com/>

Express Hints User's Guide, Version 3 Release 1

Fifth Edition (February 2013)

This edition applies to Version 3 Release 1 of the Express Hints Product, and to all subsequent releases until superseded by new editions or modified by technical documentation updates.

Reader comments on this document are welcomed and encouraged. Comments may be sent to:

HLS Technologies, Inc.
Technical Publications Group
3322 Sturbridge Lane
Sugar Land, Texas 77479-2223

© Copyright HLS Technologies, Inc. 2013. All rights reserved.

Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise without the prior written of the publisher, HLS Technologies, Inc.

Table of Contents

Preface	1
Introduction	2
Why You Need Express Hints	2
What Express Hints Does	2
Requirements	2
Restrictions	3
Installing Express Hints	4
Planning for Installation - Before You Begin.....	4
Uploading and Unpacking the Product Files	5
Running the Setup CLIST	5
Running the Installation Jobs	6
Setting Security	7
Activating Express Hints.....	7
Using Express Hints for Static SQL	8
Specifying Basic Parameters.....	8
Selecting a Package	9
Selecting an SQL Statement	9
Specifying the Access Path	9
Modifying an Access Path	10
Creating the Hint.....	12
Using Express Hints for Dynamic SQL	14
Specifying Basic Parameters.....	14
Selecting a Package	15
Selecting an SQL Statement	15
Specifying the Access Path	15
Viewing/Modifying an Access Path	16
Creating the Hint.....	18
Creating a Statement Cost Comparison	19
Using Other Features	21
Showing the Complete List of Key Columns for an Index	21

Showing Package Summary Information..... 23

Using Batch Reports..... 24

 Creating Batch Reports..... 24

 Viewing Batch Report Descriptions 26

Error Messages 29

Glossary..... 72

Index..... 75

Notices 78

Preface

This book describes how to install and use Express Hints. Express Hints is a software product that allows you to create, verify, and assign optimization hints for the purpose of controlling access path selection for SQL queries in DB2 for OS/390 and z/OS.

Who should Use this Book

This book is for data base administrators (DBA's) and application programmers who need to use the optimization hints facility to influence the determination of access paths by the DB2 optimizer. This book is also designed for DBA's or system programmers responsible for installing and supporting the Express Hints product.

Introduction

Express Hints is a powerful tool that allows DBA's and Application programmers to easily manipulate the PLAN_TABLE settings that influence the access path choices for a given SQL Statement. This section describes why you would need Express Hints and what it does in more detail.

Why You Need Express Hints

The Optimization Hints feature was introduced in DB2 Version 6. This facility allows the user to override the default access paths chosen by the DB2 optimizer. As implemented, however, the feature is far from easy to use: it requires the user to manually update the PLAN_TABLE and then rebind the plan or package just to determine whether the intended access path is even valid.

Using Optimization Hints is therefore a difficult exercise, because of the skill level required to perform the table updates successfully as well as the inherently error-prone nature of the (essentially manual) process. Even the most experienced DBA's may not have each of the many parameter settings that need to be changed at their fingertips. The percentage of application programmers possessing the requisite skill is even smaller. On top of this, the only method provided for validating a hint is to do an actual bind!

Express Hints resolves these problems.

What Express Hints Does

Express Hints gives DBA's and Application programmers the ability to easily manipulate the PLAN_TABLE settings that influence the access path choices for a given SQL Statement.

Express Hints makes it a simple matter to generate the correct PLAN_TABLE entry to direct DB2 to a desired access path. It then verifies the validity of the selected access path without forcing the user to actually bind the plan or package.

Using a straightforward ISPF dialog, the programmer or DBA selects the plan or package to be manipulated from a displayed list. The individual SQL statements are then displayed. The user selects an SQL statement and is shown the current access path setting, based on the most recent Explain data from the PLAN_TABLE. Various options are offered for changing the access path. When a new access path has been selected, it is then verified to see whether DB2 will accept it; only then is the Hint actually added to the PLAN_TABLE.

With Express Hints, changing access paths becomes a simple, direct process instead of a technical tour de force. For the first time, you can actually specify an access path for DB2 to use, instead of being at the mercy of the choices made by the Optimizer.

Requirements

Express Hints requires the following:

- DB2 Version 6 or later.

- Plan tables to be used with Express Hints must have the appropriate number of columns for the release of DB2 being accessed.

Restrictions

Express Hints supports only static SQL for DB2 versions 8 and 9.

Express hints supports both static and dynamic SQL for version 10.

In order for Express Hints to report costing information for dynamic SQL in version 10, you must have a trace running to capture IFCID 118. IFCID 118 records must be available to support cost reporting.

Installing Express Hints

This chapter will guide you through the steps needed to install Express Hints.

To install Express Hints

1. Plan for installation (see "Planning for Installation - Before You Begin" page 4).
2. Upload and unpack the product files (see "Uploading and Unpacking the Product Files" page 5).
3. Run the setup CLIST (see "Running the Setup CLIST" page 5).
4. Run the installation jobs (see "Running the Installation Jobs" page 6).
5. Set security (see "Setting Security" page 7).
6. Activate Express Hints (see "Activating Express Hints" page 7).

Planning for Installation - Before You Begin

Express Hints uses several tables. Some are standard DB2 catalog tables, and some are work tables Express Hints uses to hold intermediate results during processing. Some of the work tables are created by SETUP processing (more about SETUP later).

Express Hints uses the following principal tables:

- The user's Plan Table (userid.PLAN_TABLE). This is a standard DB2 table.
- The user's Statement Table (userid.DSN_STATEMENT_TABLE). This is a standard DB2 table.
- The user's User Query Table (userid.DSN_USERQUERY_TABLE). This is a standard DB2 table.
- The user's Express Hints History Table (userid.DSN_STATEMENT_HISTORY_TABLE). This is one of the work tables specific to Express Hints, as mentioned above. It is created by SETUP. The History Extract job updates the History Table and the Express Hints dialog reads it as input. The History Table identifies the dynamic SQL statements that are available for enhancement by optimization hints.
- The first of two Express Hints work tables (authid1.DSN_STATEMENT_CACHE_TABLE). SETUP creates the table. authid1 is an arbitrary qualifier you specify during SETUP processing. Only the History Extract job uses the qualifier so it does not need to correspond to any actual authid.
- The second Express Hints work table (authid2.DSN_STATEMENT_CACHE_TABLE). SETUP creates the table. authid2 is an arbitrary qualifier you specify during SETUP processing. Only the History Extract job uses the qualifier so it does not need to correspond to any actual authid.

Uploading and Unpacking the Product Files

Express Hints is delivered as two files named `hxhfile1.bin` and `hxhfile2.bin`. Both are required. The files are shipped in TSO Transmit format. First you must upload the two product files to your MVS system. You can use any valid file transfer program, for example, FTP, or `IND$FILE`. Once File 1 and File 2 have been uploaded to your MVS host, they must be unpacked (converted back to their original partitioned data set format.)

To upload and unpack the product files

1. Preallocate two data sets on the MVS system to receive the transferred files. Do not allow the receiving data sets to be allocated automatically by the file transfer program, as this will usually result in the files having incorrect attributes, causing subsequent installation steps to misbehave or fail altogether.
2. Allocate the data sets with the attributes **RECFM=FB, LRECL=80**, and any valid blocksize.
3. Transfer the files in binary mode.

Note: You may use any valid data set names for the two uploaded files. For the remainder of this discussion, the uploaded files will be referred to as File 1 and File 2, respectively.

4. Unpack the files using the TSO RECEIVE command. The command format is:
TSO RECEIVE INDA('data.set.name')
5. You must issue this command once for each file. In the command format in step 4, replace `data.set.name` with the actual data set names you assigned to the uploaded files. For example, if the files were named `USER55.EXHINTS.FILE1` and `USER55.EXHINTS.FILE2`, then you would issue the following commands one at a time:
 - TSO RECEIVE INDA('USER55.EXHINTS.FILE1')
 - TSO RECEIVE INDA('USER55.EXHINTS.FILE2')
6. Each time you issue the RECEIVE command, message **INMR906A** appears. Do one of the following:
 - Press ENTER, which will cause the reconstituted PDS library to be created using your TSO ID as the high level qualifier
 - Enter the subcommand **DA('data.set.name')**, which will generate the PDS using the name you specified.

Note: Remember the names assigned to the unpacked files. You will need them for the next installation step.

Running the Setup CLIST

The next step for installation is to run the setup CLIST.

To Run the Setup CLIST

1. Once you successfully execute the receive commands, you will have a pair of PDS libraries corresponding to File 1 and File 2.

File 1 contains a CLIST named SETUP that creates and initializes the actual product data sets. Execute this CLIST, either from the library where it resides or after copying it to a CLIST library of your choice. You will be prompted to enter the following values, in the order shown:

- The data set name of the first installation file (the first file RECEIVED and unpacked).
- The data set name of the second installation file (the loadlib RECEIVED and unpacked).
- High level qualifier(s) for prefixing the product data set names.
- The name of an existing data base to contain the tables required by the cache extract program (HXH100).
- A creator ID for the History Table (output from the extract program).
- A creator ID for the first of two work tables needed by the extract program.
- A creator ID for the second work table.

Note: The two work tables must have different creator IDs. Do not use the same ID for both. The Express Hints dialog will not give usable results if you do this.

- The owner ID to be used in the BIND commands for installation of the Express Hints programs.
- The qualifier ID to be used in the BIND commands for installation of the Express Hints programs.

SETUP will create and populate a series of data sets containing the product elements. One of these data sets has a name ending in **CONTROL**. This is a sequential data set consisting of one record, the date key. For any Express Hints program to execute successfully, the CONTROL file must contain a valid date key.

2. You should have received a date key from HLS as part of the product delivery. Using ISPF Edit, insert the supplied date key in column 1 of line 1 of the CONTROL file.
3. Once the two installation files have been uploaded and received, you are ready to run the installation job(s).

Running the Installation Jobs

This section describes the procedure for running the installation jobs.

To Run the Installation Jobs

1. Run the SAMPLIB jobs in the following order. Each job contains comments explaining what to change before submitting the job. Jobs are listed by their member names in SAMPLIB.
 - BIND
 - TABLES
 - EXTRACT

The product is now fully installed.

2. To use the Express Hints dialog, execute the CLIST named EXHINTS. This is located in the product ISPCLIB. You can execute it directly from there, or copy it to your own CLIST library and execute from there. To execute the CLIST from the installed product library (ISPCLIB), issue a command like the following, where `ispclib_dataset_name` is the fully qualified data name of ISPCLIB.

TSO EX '`ispclib_dataset_name(EXHINTS)`'

Note: Express Hints is an ISPF dialog and must be run under ISPF. It will not function correctly if executed from a TSO READY prompt.

Setting Security

After you successfully run SETUP, the libraries it creates will contain the executable Express Hints product components. The users of Express Hints will require READ access to those libraries in order to use the product. Have your security administrator grant the appropriate level of access to the libraries created by SETUP to all users who need to run Express Hints.

Activating Express Hints

The systems programmers at your installation may have arranged another way for you to invoke the program. In this case they will tell you what it is. To use the default activation, do the following procedure.

To Activate Express Hints

- In the **command** field of any ISPF panel, type **TSO EXHINTS**.
Express Hints appears with the option to choose between Dynamic and Static SQL.

Using Express Hints for Static SQL

DB2's optimization hints facility is designed to let you insert a row in your PLAN_TABLE with the access path and hint name you specify, so that you can use that hint when rebinding the package. In this way, you can direct DB2 to use the access path you request and not one of its own choosing.

This chapter describes the procedures for using Express Hints for Static SQL.

To Use Express Hints for Static SQL

1. Specify Basic Parameters (see "Specifying Basic Parameters" page 8).
2. Select a Package (see "Selecting a Package" page 15).
3. Select an SQL Statement (see "Selecting an SQL Statement" page 15).
4. Specify the Access Path (see "Specifying the Access Path" page 9).

You can also use other optional features of Express Hints (see "Using Other Features" page 21).

Specifying Basic Parameters

This section describes how to specify the basic parameters in Express Hints.

To Specify Basic Parameters

1. Activate Express Hints (see "Activating Express Hints" page 7).
2. In the **Option** field, type 1 for STATIC.
3. Press ENTER.

The Express Hints Main panel appears.

4. Enter the following values:
 - **Subsystem ID:** The subsystem ID (SSID) of the DB2 subsystem containing the packages you want to manipulate.
 - **PLAN_TABLE owners Source table:** The user ID (authid) of the owner of the source PLAN_TABLE. This is the PLAN_TABLE containing the Explain data for your packages.
 - **DBRM/package name:** The name of the DBRM or package that contains the SQL statement(s) whose access paths you want to manipulate.
5. If you want to specify the destination PLAN_TABLE, enter the following value:
 - **PLAN_TABLE owners Destination table:** The user ID (authid) of the owner of the destination PLAN_TABLE. This is the PLAN_TABLE where the new Hint you create will be stored. If you leave this value blank, new Hints are stored in the source PLAN_TABLE.
6. Press ENTER.

Express Hints searches the DB2 catalog for all occurrences of the specified DBRM. The results of the search appear in a list.

Selecting a Package

Before you can select a package, you must run Express Hints and search the DB2 catalog (see "Specifying Basic Parameters" page 8) for the packages containing the DBRM you want to use. The Item Selection panel lists the packages that contain the specified DBRM allowing you to select one for detailed processing.

The **Type** column shows **PKG** for package. The **Collection** column lists the names of the different packages.

To Select a Package

1. To the left of the package you want to select, type **s** for select.
2. Press ENTER.

Express Hints checks for how many versions of that package exist in the DB2 subsystem. If multiple versions exist, a list of the available versions appears so you can specify which one to process.

3. If multiple versions exist, do the following:
 - a. Under **Version**, type **s** to the left of the version you want to process.
 - b. Press ENTER.

Express Hints retrieves the SQL statements for the specified DBRM. The statements appear in a list.

Selecting an SQL Statement

After you have selected a package(see "Selecting a Package" page 15), Express Hints retrieves the SQL statements for the specified DBRM. The statements appear in a list under the **header Choose a statement to process**.

The **Queryno** column contains the ID number of the statement, and the **Statement** column contains the description of the statement.

To Select an SQL Statement

1. To the left of the statement you want to process, type **s**.
2. Press ENTER.

The Access Path Control submenu appears.

Specifying the Access Path

After you have selected a statement (see "Selecting an SQL Statement" page 9), the Access Path Control submenu appears. The Access Path Control submenu contains the following three options:

- **Modify:** Change the access path.

- **Name:** Assign a hint name.
- **Make:** Create a hint.

You can do the following:

- Modify an Access Path (see "Modifying an Access Path" page 10).
- Create the Hint (see "Creating the Hint" page 12).

Modifying an Access Path

From the Access Path Control submenu (see "Specifying the Access Path" page 9), the MODIFY command allows you to specify the access path you want to use.

To Modify an Access Path

1. In the **COMMAND** field, type **1** for MODIFY.
2. Press ENTER

The Query Elements panel appears. This panel lists the query elements that comprise the selected SQL statement. If the statement involves a join, multiple query elements will be listed. If not, there will be only one.

If multiple query elements exist, they are listed in the order in which DB2 uses them to perform the join (the join sequence).

3. If you want to change the Join Sequence, then to the left of the **Sequence** column, type new sequence numbers and press ENTER. Example:

```

Menu utilities options Help
-----
Express Hints - Query Elements          Row 1 to 2 of 2
COMMAND ==>                             Scroll ==> CSR
Subsystem . . . . . : DSN1
DBRM . . . . . : DBRM01
Collection . . . . . : DBRPK01
Version . . . . . : VERSION15
Queryno . . . . . : 127

Choose a query element to edit, or specify a different sequence

Sequence  Table Accessed
-----
2__      001      USER01.USER_BD
1__      002      USER01.GROUP_BD
***** Bottom of data *****

```

The Query Elements panel reappears with the changed sequence.

4. To the left of the sequence you want to select, type **s** for select.
5. Press ENTER.

The Access Path Settings panel appears. This panel shows the current access path settings for the selected query element.

```

Menu Utilities Options Help
-----
Express Hints - Access Path Settings
COMMAND ==>
Subsystem . : DSN1          Package . . : DBRM01          Query No. . : 1279
-----
Prefetch mode (PREFETCH)
4  1. S      Sequential prefetch
   2. L      List prefetch
   3. D      Dynamic
   4. Blank  (no prefetch)

Type of Access (ACCESSTYPE)
1  1. I      By the index named below
   2. I1     1-fetch index scan
   3. N      Index scan with 'IN' predicate
   4. R      Table space scan
   5. M      Multiple index scan
   6. Blank  Not applicable

Index . . . . . USER01.GROUP_BD_IX1

Number of index keys (MATCHCOLS) . . . . 1

Enter "/" to select option
/  Data access not required (INDEXONLY)
   Attempt to use direct row access (PRIMARY_ACCESSTYPE)

```

6. Do any of the following:

If you want to	Then
Change the prefetch mode	<p>Under Prefetch mode, delete the existing selection and type a new selection.</p> <p>In the previous example, you would delete 4 (for Blank - no prefetch), and type 1, 2, or 3.</p>
Change the type of access	<p>Under Type of Access, delete the existing selection and type a new selection.</p> <p>In the previous example, you would delete 1 (for I - By the index named below), and type any number between 2 and 6.</p>
Specify a different index to use and you know the name of the index	<p>In the Index field, type the name of the new index.</p>
Search for a list of available indexes to select a new index	<ol style="list-style-type: none"> In the Index field, type ?. Press ENTER. <p>The Available Indexes panel appears with the list of indexes. The index information is shown one index per line. Each line contains the name of the index as well as the list of key columns, in order</p>

If you want to	Then
	<p>from major to minor, with ascending/descending sequence indicators.</p> <div data-bbox="600 363 1409 552" style="border: 1px solid black; background-color: #ffffcc; padding: 5px;"> <p>Note: Because the space available for listing index keys is limited, the complete list of keys for an index may not fit on one line. If the number of keys is too large to fit the display, there is a way to display the complete list (see "Showing the Complete List of Key Columns for an Index" page 21).</p> </div> <p>c. To the left of the index you want to select, type /.</p> <p>d. Press ENTER.</p> <p>The Access Path Settings panel reappears with the new index selected.</p>

7. Press ENTER and the system validates the changes.
8. Press F3 to return to the Access Path Control submenu.

Creating the Hint

From the Access Path Control submenu (see "Specifying the Access Path" page 9), you can create the hint. You will have to use both the NAME and MAKE commands to first name, and then create the hint.

To Create the Hint

1. In the **COMMAND** field, type **2** for NAME.
2. Press ENTER.

The Hint Name panel appears. The existing hint name (if any) appears on the left, and an entry field on the right allows you to specify the new hint name.

3. In the **New hint name** field, type the name of the hint.
4. Press F3.

The Access Path Control Submenu reappears.

5. In the **COMMAND** field, type **3** for MAKE.
6. Press ENTER.

Express Hints dynamically invokes DB2 to determine whether the hint is acceptable (whether the access path is logically valid). If the hint is acceptable, a confirmation message appears.


```

Menu utilities options Help
-----
Express Hints - Access Path Control
COMMAND ==>
Subs
DBRM
Coll
Vers
Quer
Sele
      COMMAND ==>
      Confirm Hint
      This hint has been validated and accepted by
      DB2. If you wish, the hint can be saved in
      the plan table so that it may be used to bind
      with in the future.
      Press ENTER to save the hint in the plan
      table.
      Press END to return to the previous panel
      without saving the hint.

```

7. Do one of the following:

If you want to	Then
Quit without saving	Press END.
Update the PLAN_TABLE with the new hint	Press ENTER. Express Hints inserts a row in the PLAN_TABLE containing the access path and hint name you specified, and a confirmation message appears.

Once your hint has been added to the PLAN_TABLE, you can then bind the package, specifying the new hint name on the BIND command. This will cause DB2 to use the access path you have specified.

Using Express Hints for Dynamic SQL

DB2's optimization hints facility allows you to insert a row in your PLAN_TABLE with the access path and hint name you specify, so that you can use that hint when rebinding the package. In this way, you can direct DB2 to use the access path you request and not one of its own choosing.

This chapter describes the procedures for using Express Hints for Dynamic SQL.

To Use Express Hints for Dynamic SQL

1. Specify Basic Parameters (see "Specifying Basic Parameters" page 14).
2. Select a Package (see "Selecting a Package" page 9).
3. Select an SQL Statement (see "Selecting an SQL Statement" page 15).
4. Specify the Access Path (see "Specifying the Access Path" page 15).

Specifying Basic Parameters

This section describes how to specify the basic parameters in Express Hints.

To Specify Basic Parameters

1. Activate Express Hints (see "Activating Express Hints" page 7).
2. In the **Option** field, type **2** for DYNAMIC.
3. Press ENTER.

The Express Hints Main panel appears.

4. In the **Select the SQL statement to be tuned** section, enter the following values:
 - **Subsystem ID:** The subsystem ID (SSID) of the DB2 subsystem containing the packages you want to manipulate.
 - **Owner ID for EXPLAIN tables:** The prefix for the EXPLAIN tables you want to use.
 - **Owner ID for History table:** The owner ID that will prefix the DSN_STATEMENT_HISTORY_TABLE. This table is used to identify and locate the statements.
5. In the SQL statement selection (filter) criteria section, enter the following values:

Note: The following three values are filters Express Hints uses to select a list of SQL statements

- **Schema:** The schema ID or prefix.
 - **Collection.** The collection ID.
 - **Program:** The DBRM package name.
6. Press ENTER.

Express Hints searches the DB2 catalog for all the SQL statements available for processing. The results of the search list the statements based on the filters specified.

Selecting a Package

Before you can select a package, you must run Express Hints and search the DB2 catalog (see "Specifying Basic Parameters" page 8) for the packages containing the DBRM you want to use. The Item Selection panel lists the packages that contain the specified DBRM allowing you to select one for detailed processing.

The **Type** column shows **PKG** for package. The **Collection** column lists the names of the different packages.

To Select a Package

1. To the left of the package you want to select, type **s** for select.
2. Press ENTER.

Express Hints checks for how many versions of that package exist in the DB2 subsystem. If multiple versions exist, a list of the available versions appears so you can specify which one to process.

3. If multiple versions exist, do the following:
 - c. Under **Version**, type **s** to the left of the version you want to process.
 - d. Press ENTER.

Express Hints retrieves the SQL statements for the specified DBRM. The statements appear in a list.

Selecting an SQL Statement

After you have selected a package (see "Selecting a Package" page 15), Express Hints retrieves the SQL statements for the specified DBRM. The statements appear in a list under the **header Choose a statement to process**.

The **Queryno** column contains the ID number of the statement, and the **Statement** column contains the description of the statement.

To Select an SQL Statement

1. To the left of the statement you want to process, type **s**.
2. Press ENTER.

The Access Path Control submenu appears.

Specifying the Access Path

After you have selected a statement (see "Selecting an SQL Statement" page 15), the Access Path Control submenu appears. The Access Path Control submenu contains the following four options:

- **Display/Modify:** View or change the access path.
- **Name:** Assign a hint name.
- **Make:** Create a hint.

- **Cost:** Create a statement cost comparison.

You can do the following:

- View/Modify an Access Path (see "Viewing/Modifying an Access Path" page 16).
- Create the Hint (see "Creating the Hint" page 18).
- Create a Statement Cost Comparison (see "Creating a Statement Cost Comparison" page 19).

Viewing/Modifying an Access Path

From the Access Path Control submenu (see "Specifying the Access Path" page 15), the DISPLAY/MODIFY option allows you to view and change an access path.

To View/Modify an Access Path

1. In the **COMMAND** field, type **1** for DISPLAY/MODIFY.
2. Press ENTER.

The Query Elements panel appears. This panel lists the query elements that comprise the selected SQL statement. If the statement involves a join, multiple query elements will be listed. If not, there will be only one.

If multiple query elements exist, they are listed in the order in which DB2 uses them to perform the join (the join sequence).

3. If you want to change the Join Sequence, then to the left of the **Sequence** column, type new sequence numbers and press ENTER. Example:

```

Menu utilities options Help
-----
Express Hints - Query Elements          Row 1 to 2 of 2
COMMAND ==>                             Scroll ==> CSR
Subsystem . . . . . : DSN1
DBRM . . . . . : DBRM01
Collection . . . . . : DBRPK01
Version . . . . . : VERSION15
Queryno . . . . . : 127

Choose a query element to edit, or specify a different sequence

Sequence  Table Accessed
-----
2__      001      USER01.USER_BD
1__      002      USER01.GROUP_BD
***** Bottom of data *****

```

The Query Elements panel reappears with the changed sequence.

4. To the left of the sequence you want to select, type **s** for select.
5. Press ENTER.

The Access Path Settings panel appears. This panel shows the current access path settings for the selected query element.

```

Menu utilities options help
-----
* Express Hints for DB2 *
COMMAND ==>
-----
Prefetch mode (PREFETCH) More: +
1 1. S Sequential prefetch
   2. L List prefetch
   3. D Dynamic
   4. Blank (no prefetch)

Type of Access (ACCESSTYPE)
1 1. I By the index named below
   2. I1 1-fetch index scan
   3. N Index scan with 'IN' predicate
   4. R Table space scan
   5. M Multiple index scan
   6. Blank Not applicable

Index . . . . . WNFRI.PLAN_TABLE_HINT_IX
Number of index keys (MATCHCOLS) . . . . 0

Enter "/" to select option
Data access not required (INDEXONLY)
Attempt to use direct row access (PRIMARY_ACCESSTYPE)

Join method for this step (METHOD)
0 0. First table accessed, continuation of previous, or not used
1 1. Nested loop join

```

6. Do any of the following:

If you want to	Then
Change the prefetch mode	<p>Under Prefetch mode, delete the existing selection and type a new selection.</p> <p>In the previous example, you would delete 1 (for S - Sequential prefetch), and type any number between 2 and 4.</p>
Change the type of access	<p>Under Type of Access, delete the existing selection and type a new selection.</p> <p>In the previous example, you would delete 1 (for I - By the index named below), and type any number between 2 and 6.</p>
Specify a different index to use and you know the name of the index	<p>In the Index field, type the name of the new index.</p>
Search for a list of available indexes to select a new index	<p>a. In the Index field, type ?.</p> <p>b. Press ENTER.</p> <p>The Available Indexes panel appears with the list of indexes. The</p>

If you want to	Then
	<p>index information is shown one index per line. Each line contains the name of the index as well as the list of key columns, in order from major to minor, with ascending/descending sequence indicators.</p> <div data-bbox="602 436 1398 625" style="border: 1px solid black; background-color: #ffffcc; padding: 5px;"> <p>Note: Because the space available for listing index keys is limited, the complete list of keys for an index may not fit on one line. If the number of keys is too large to fit the display, there is a way to display the complete list(see "Showing the Complete List of Key Columns for an Index" page 21).</p> </div> <p>c. To the left of the index you want to select, type /.</p> <p>d. Press ENTER.</p> <p>The Access Path Settings panel reappears with the new index selected.</p>

7. Press ENTER and the system validates the changes.
8. Press F3 to return to the Access Path Control submenu.

Creating the Hint

From the Access Path Control submenu (see "Specifying the Access Path" page 15), you can create the hint. You will have to use both the NAME and MAKE commands to first name, and then create the hint.

To Create the Hint

1. In the **COMMAND** field, type **2** for NAME.
2. Press ENTER.

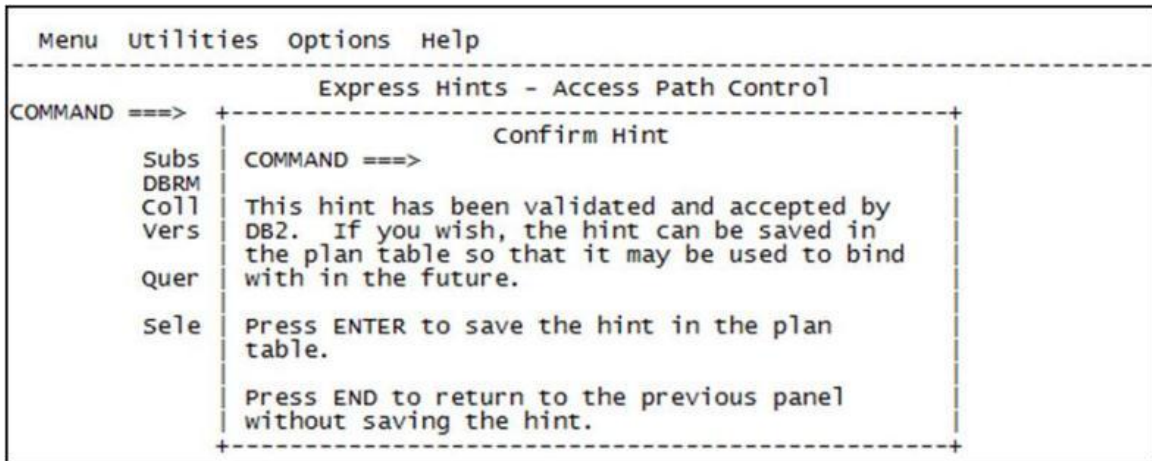
The Hint Name panel appears. The existing hint name (if any) appears on the left, and an entry field on the right allows you to specify the new hint name.

3. In the **New hint name** field, type the hint name.
4. Press F3.

The Access Path Control submenu appears.

5. In the **COMMAND** field, type **3** for MAKE.
6. Press ENTER.

Express Hints dynamically invokes DB2 to determine whether the hint is acceptable (whether the access path is logically valid). If the hint is acceptable, a confirmation message appears.



7. Do one of the following:

If you want to	Then
Quit without saving	Press END.
Update the PLAN_TABLE with the new hint	Press ENTER. Express Hints inserts a row in the PLAN_TABLE containing the access path and hint name you specified, and a confirmation message appears.

Once your hint has been added to the PLAN_TABLE, you can then bind the package, specifying the new hint name on the BIND command. This will cause DB2 to use the access path you have specified.

Creating a Statement Cost Comparison

Statement cost comparison allows you to display an estimated cost of executing the statement both with and without using the hint. This tool helps you to decide if you want to proceed using the hint as well as highlighting the cost savings of using the hints you create. You can create a statement cost comparison from the Access Path Control submenu(see "Specifying the Access Path" page 15).

To Create a Statement Cost Comparison

1. In the **COMMAND** field, type **4** for COST.
2. Press ENTER.

The Cost Comparison panel appears in one of two different formats depending on where you are in the process.

- Cost Comparison panel without a created hint.

```

Menu utilities Options Help
-----
* Express Hints for DB2 *
COMMAND ==> 4
+----- Estimated Statement Costs -----+
| Menu utilities Options Help |
|                               |
| COMMAND ==>                  |
| original access path         |
|                               |
| Milliseconds . . . . . : 122 |
| Service Units . . . . . : 238 |
|                               |
| Total Cost . . . . . : 1450.8897 |
+-----+

```

- Cost Comparison panel with a hint.

```

Menu utilities Options Help
-----
* Express Hints for DB2 *
COMMAND ==> 4
+----- Estimated Statement Costs -----+
| Menu utilities Options Help |
|                               |
| COMMAND ==>                  |
| original access path         |
|                               |
| Milliseconds . . . . . : 122 |
| Service Units . . . . . : 238 |
|                               |
| Total Cost . . . . . : 1450.889 |
|                               |
| Modified access path        |
|                               |
| Milliseconds . . . . . : 52  |
| Service Units . . . . . : 100 |
|                               |
| Total Cost . . . . . : 148.4996 |
+-----+

```


Using Other Features

This section briefly describes several optional features of Express Hints.

You can do the following:

- Show the Complete List of Key Columns for an Index (see "Showing the Complete List of Key Columns for an Index" page 21).
- Show Package Summary Information (see "Showing Package Summary Information" page 23).

Showing the Complete List of Key Columns for an Index

When you are modifying an access path, you can search for an index you want to specify. The search results cannot always show all the key columns for an index. You can use the procedure in this section to show the complete list of key columns to help you select the index you want to use.

To Show the Complete List of Key Columns for an Index

1. Complete steps 1 through 5 of Modifying an Access Path (for Static SQL) or Viewing/Modifying an Access Path (for Dynamic SQL)(see "Modifying an Access Path" page 10) or (see "Viewing/Modifying an Access Path" page 16).

The Access Path Settings panel appears. This panel shows the current access path settings for the selected query element.

```
Menu utilities options Help
-----
Express Hints - Access Path Settings
COMMAND ==>
Subsystem . : DSN1          Package . . : DBRM01          Query No. . : 1279
-----
More: +
Prefetch mode (PREFETCH)
4  1. S Sequential prefetch
   2. L List prefetch
   3. D Dynamic
   4. Blank (no prefetch)
Type of Access (ACCESSTYPE)
1  1. I By the index named below
   2. I1 1-fetch index scan
   3. N Index scan with 'IN' predicate
   4. R Table space scan
   5. M Multiple index scan
   6. Blank Not applicable
Index . . . . . USER01.GROUP_BD_IX1
Number of index keys (MATCHCOLS) . . . . 1
Enter "/" to select option
/ Data access not required (INDEXONLY)
  Attempt to use direct row access (PRIMARY_ACCESSTYPE)
```

2. In the **Index** field, type ?.
3. Press ENTER.

The list of available indexes appears. In the example, the last index **DS_BD_IX99** has more key columns than will fit on the display. The plus sign (+) at the end of the line indicates that there is more data than can be shown on the screen.

```

Menu Utilities Options Help
----- Available Indexes -----
Row 1 to 7 of 7
COMMAND ===> Scroll ===> CSR

The following indexes are defined for this table

----- Index Name ----- Index Key Columns -----
. USER01.DS_BD_IX1          DSBD_NAME (A), DSBD_VOL (A)
. USER01.DS_BD_IX2          DSBD_OWNER_ID (A)
. USER01.DS_BD_IX3          DSBD_GRP_ID (A)
. USER01.DS_BD_IX4          DSBD_SECLEVEL (A)
. USER01.DS_BD_IX5          DSBD_NOTIFY_ID (A)
. USER01.DS_BD_IX6          DSBD_SECLABEL (A)
. USER01.DS_BD_IX99         DSBD_NAME (A), DSBD_VOL (A), DSBD_OWNER+
***** Bottom of data *****

```

4. To the left of the index for which you want to see columns, type **c** for columns.
5. Press ENTER.

A scrollable popup window appears containing the list of key columns for the index. The columns are listed in order from major to minor. Each key column name is listed under the **Column** heading, with a corresponding entry in the **Order** column indicating whether the keys in that column are accessed in ascending (**A**) or descending (**D**) order.

```

Menu Utilities Options Help
----- Available Indexes -----
Row 1 to 7 of 7
COMMAND ===> Scroll ===> CSR

----- Index Name ----- Index Key Columns -----
. USER01.DS_BD_IX1          DSBD_NAME (A), DSBD_VOL (A)
. USER01.DS_BD_IX2          DSBD_OWNER_ID (A)
. USER01.DS_BD_IX3          DSBD_GRP_ID (A)
. USER01.DS_BD_IX4          DSBD_SECLEVEL (A)
. USER01.DS_BD_IX5          DSBD_NOTIFY_ID (A)
. USER01.DS_BD_IX99         DSBD_NAME (A), DSBD_VOL (A), DSBD_OWNER+
***** Bottom of data *****

COMMAND ===>
Row 1 to 5 of 5
Scroll ===> CSR

Index key columns

Seq  Order  Column
1    A      DSBD_NAME
2    A      DSBD_VOL
3    A      DSBD_OWNER_ID
4    A      DSBD_DEVICE_NAME
5    A      DSBD_NOTIFY_ID
***** Bottom of data *****

DSBD_OWNER+
*****

```

Showing Package Summary Information

After you search the DB2 catalog for the packages containing the DBRM you want to use.

For Static SQL (see "Specifying Basic Parameters" page 8).

For Dynamic SQL (see "Specifying Basic Parameters" page 14).

The Item Selection panel lists the packages that contain the specified DBRM allowing you to view the package summary information.

The **Type** column shows **PKG** for package. The **Collection** column lists the names of the different packages.

To Show Package Summary Information

1. To the left of the package you want to view, type **i** for information.
2. Press ENTER.

The DBRM - Package (Detail) panel appears with the package summary information.

Using Batch Reports

Express Hints allows you to capture the EXPLAIN data (access path information) for SQL statements extracted from the dynamic statement cache. By doing this at two different points in time, you can then generate a series of reports comparing the two sets of access paths and highlighting key differences. The data can be from the same subsystem, or from two different subsystems.

The reports contrast the access paths and their associated costs, as calculated by the Optimizer. You can use this information for migrations, for quality assurance, or for general tuning.

To create the reports, you must first extract the dynamic SQL from the cache by using program HXH100, the History Extract program. The output from HXH100 is the principal input to the reporting facility.

Note: Batch Reports are currently only supported by Dynamic SQL.

You can do the following:

- Create Batch Reports (see "Creating Batch Reports" page 24).
- View Batch Report Descriptions (see "Viewing Batch Report Descriptions" page 26).

Creating Batch Reports

This section describes the procedure for creating batch reports.

To Create Batch Reports

1. Run the first history extract **HXH100**.
2. Run the second history extract **HXH100** (from the same or a different subsystem).
3. Run the access path capture program **HXH201** against the output of step 1.
4. Run the access path capture program **HXH201** against the output of step 2.
5. Generate the reports using **HXH202**, the report program.

Sample JCL

JCL for HXH201 and HXH202 is in the product SAMPLIB. See the following examples of typical execution JCL.

Example HXH201

```

//REPEX  PROC  NEWDSN=,
//
//REPEX1 EXEC  PGM=IEFBR14
//EXOUT      DD  DISP=(MOD,DELETE),
//
//              UNIT=SYSALLDA,
//              SPACE=(TRK,1,RLSE),
//              DSORG=PS,RECFM=FB,LRECL=32447,
//              DSN=&NEWDSN
//*
//REPEX2 EXEC  PGM=HXH201
//STEPLIB    DD  DISP=SHR,DSN=DNS1.HXHV3R2.LOADLIB
//
//              DD  DISP=SHR,DSN=DSNA10.SDSNLOAD
//CONTROL    DD  DISP=SHR,DSN=DNS1.HXHV3R2.CONTROL
//SYSPRINT   DD  SYSOUT=*
//EXOUT      DD  DISP=(MOD,CATLG),
//
//              UNIT=SYSALLDA,
//              SPACE=(TRK,&SIZE,RLSE),
//              DSORG=PS,RECFM=FB,LRECL=32447,
//              DSN=&NEWDSN
//
//          PEND
//RUN        EXEC  REPEX,NEWDSN=DNS1.EXOUTA
//REPEX2.SYSIN  DD  *
SUBSYSTEM DBAG
HISTORYOWNER DNS1
EXPLAINOWNER DNS1
/*

```

Example HXH202

```

//GENERATE EXEC PGM=HXH202
//STEPLIB DD DISP=SHR,DSN=DNS1.HXHV3R2.LOADLIB
// DD DISP=SHR,DSN=DSNA10.SDSNLOAD
//CONTROL DD DISP=SHR,DSN=DNS1.HXHV3R2.CONTROL
//SYSPRINT DD SYSOUT=*
//REPORT1 DD SYSOUT=*
//REPORT2 DD SYSOUT=*
//REPORT3 DD SYSOUT=*
//REPORT4 DD SYSOUT=*
//REPORT5 DD SYSOUT=*
//OLD DD DISP=SHR,DSN=DNS1.OLD.PATHDATA
//NEW DD DISP=SHR,DSN=DNS1.NEW.PATHDATA
//SYSIN DD *
SUBSYSTEM <SSID>
/*

```

Viewing Batch Report Descriptions

The HXH202 program produces up to five reports. Each report is generated using a different DD statement (REPORT1 through REPORT5). To generate a particular report, simply include the associated DD statement in the execution JCL. The program only generates reports for the DD statements present.

- Report 1 – Lists all SQL statements that match. For example, those that are present in both the “new” and “old” inputs.
- Report 2 – Lists only the SQL statements whose access paths have changed.
- Report 3 – Lists only the SQL statements whose estimated CPU cost has increased. (Sorted in descending order by the amount of difference. The SQL statements with the largest difference are listed at the beginning of the report.)
- Report 4 – Lists only the SQL statements whose estimated service units have increased. (Sorted in descending order by the amount of difference. The SQL statements with the largest difference are listed at the beginning of the report.)
- Report 5 – Lists only the SQL statements whose “total cost” has increased.* (Sorted in descending order by the amount of difference. The SQL statements with the largest difference are listed at the beginning of the report.)

* *Total Cost* is a calculated value produced by the Optimizer. It does not correlate precisely to any directly measurable external metric, but is useful for delta comparisons.

All five reports have essentially the same format; only the contents differ.

A sample from Report 3.

```
=====
-- STATEMENT WITH INCREASED COST (CPU) ----- 22/22/2012  21:48:30 -----
=====
```

Subsystem DB9G (Version 9)		Subsystem DBAG (Version 10)	
Milliseconds	70	Milliseconds	2713 *** +2643 (3776%) ***
Service units	136	Service units	5288
Total Cost	66.09	Total Cost	1624.26

SQL statement:

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7
SELECT * FROM SYSIBM.SYSPACKAGE
```

```
=====
-- STATEMENT WITH INCREASED COST (CPU) ----- 22/22/2012  21:48:30 -----
=====
```

Subsystem DB9G (Version 9)		Subsystem DBAG (Version 10)	
Milliseconds	141	Milliseconds	2684 *** +2543 (1804%) ***
Service units	274	Service units	5231
Total Cost	1110.17	Total Cost	1606.30

--> Type of access has changed. Was I ,
is now R .

SQL statement:

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7
SELECT * FROM SYSIBM.SYSTABLES
```

```
=====
-- STATEMENT WITH INCREASED COST (CPU) ----- 22/22/2012  21:48:30 -----
=====
```

Subsystem DB9G (Version 9)		Subsystem DBAG (Version 10)	
Milliseconds	21	Milliseconds	2300 *** +2279 (10852%) ***
Service units	40	Service units	4483
Total Cost	152.86	Total Cost	1372.89

--> Type of access has changed. Was I ,
is now R .

SQL statement:

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7
SELECT * FROM SYSIBM.SYSPLAN
```

```
=====
-- STATEMENT WITH INCREASED COST (CPU) ----- 22/22/2012  21:48:30 -----
=====
```

Subsystem DB9G (Version 9)		Subsystem DBAG (Version 10)	
Milliseconds	4	Milliseconds	1510 *** +1506 (37650%) ***
Service units	8	Service units	2943
Total Cost	23.00	Total Cost	915.09

--> Type of access has changed. Was I ,
is now R .

SQL statement:

....+....1....+....2....+....3....+....4....+....5....+....6....+....7
SELECT * FROM SYSIBM.SYSDATABASE WHERE CREATEDBY 'SYSIBM'

Error Messages

Express Hints messages are of the form HXH*nnnt*, where *nnn* is the message number and *t* is a one-character suffix indicating the type and severity of the message. The suffix values and their meanings are as follows:

I	(Informational)	Routine notification; no action required.
W	(Warning)	Notifies the user of an anomalous circumstance that may or may not reflect an error.
E	(Error)	Processing cannot continue for the reason given in the message; probable user error.
S	(Severe)	Environmental problem or logic error; continued execution is impossible.

Each message is listed below along with its corresponding explanation.

HXH000E ***** Express Hints trial expired *mmm dd, yyyy*. Contact HLS. *****

Explanation: Self explanatory. The trial period for Express Hints ended on the date shown in the message.

User Action: Contact HLS.

HXH001E **Express Hints failed during initialization, symptom=*nnn***

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support.

User Action: Contact Technical Support.

HXH003E **Unable to connect to subsystem *ssss***

Explanation: The subsystem is not available or not responding.

User Action: Make sure the subsystem name specified on the panel is correct, and that the subsystem is up.

HXH009E **TBCREATE failed, rc=*nnn***

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support.

User Action: Contact Technical Support.

HXH013E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH014E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH015E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH017E TBADD failed for package entry, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support.

User Action: Contact Technical Support.

HXH018E No matching catalog entries were found

Explanation: No packages using the named package are known to the specified DB2 subsystem.

User Action: Specify the correct package and/or subsystem name(s) and retry.

HXH019E Too many selections (only 1 item may be processed at a time)

Explanation: Only one package may be processed at a time. You attempted to select more than one entry from the displayed list.

User Action: Select a single entry and retry.

HXH020E Invalid selection. The only valid selection characters are '/', 'S', and 'I'

Explanation: An invalid selection character was entered.

User Action: Retry using one of the selection characters listed.

HXH032E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.??

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH033E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH034E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH0353 TBCREATE failed, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support.

User Action: Contact Technical Support.

HXH036E TBADD failed for version table, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support.

User Action: Contact Technical Support.

HXH037E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH038E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, nnn is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH039E Too many selections (only one version may be processed at a time)

Explanation: Only one version of a package may be processed at a time. You attempted to select more than one entry from the displayed list.

User Action: Select a single version and retry.

HXH040E Invalid selection. The only valid selection characters are '/' and 'S'

Explanation: An invalid selection character was entered.

User Action: Retry using one of the selection characters listed.

HXH042E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, nnn is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH050E *tablename* does not exist in *ssid*

Explanation: The specified plan table does not exist in the indicated subsystem.

User Action: Make sure that the correct user ID was entered. If the user ID is correct, then there is no plan table for that user. You will need to create and populate the plan table before using Express Hints. See the DB2 manuals for detailed information on creating a plan table.

HXH051E 'PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, nnn is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH054E 'xxxx' is not a valid command

Explanation: The indicated command is not valid on the current panel.

User Action: Enter a valid command or blanks (as appropriate) and retry.

HXH070E *tablename* is not usable -- Express Hints requires at least a 58-column plan table

Explanation: Plan tables can be created in various formats with different numbers of columns. In order to use Express Hints, your plan table must have 58 columns or more; plan tables with fewer than 58 columns are not supported. In the message text, *tablename* is the name of the unusable plan table.

User Action: (1) Recreate the plan table with at least 58 columns and repopulate it with Explain output, or (2) use a different plan table.

HXH077E A hint name must be specified

Explanation: Each optimization hint requires an assigned name.

User Action: Enter a valid 1 to 8 character hint name.

HXH082E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH083E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH084E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH86I *uqtname* was created in *dbname.tsname*

Explanation: Informational message. The userquery table named *uqtname* was successfully created in table space *tsname* in database *dbname*.

User Action: None.

HXH093E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH094E **INSERT failed, sqlcode** *nnn*

Explanation: The attempt to add a row to a plan table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH098E **COMMIT failed, sqlcode** *nnn*

Explanation: A COMMIT command was issued but failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH110E **PREPARE failed, sqlcode** *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH114E **TBCREATE failed, rc=***nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support.

User Action: Contact Technical Support.

HXH115E **OPEN CURSOR failed, sqlcode** *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH116E **TBADD failed for package entry, rc=***nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support.

User Action: Contact Technical Support.

HXH117E **FETCH failed, sqlcode** *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH118E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH119E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH120E TBCREATE failed, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH121E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH122E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH123E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH126E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH127E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH128E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH129E TBADD failed for package entry, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH130E No current SQL statements for this package were found in the catalog

Explanation: Self-explanatory. The DB2 catalog does not contain any bound statement information for this package.

User Action: Verify that the application contains executable SQL statements. If this is the case, rebind the package and retry. If this fails, contact HLS Technical Support.

HXH131E Too many selections (only 1 statement may be processed at a time)

Explanation: Only one SQL statement may be processed at a time. You attempted to select more than one entry from the displayed list.

User Action: Select a single statement and retry.

HXH132E Zero or negative QUERYNO, unable to process

Explanation: The selected statement is not explainable and is therefore not subject to the optimization hints facility.

User Action: Select a different statement to hint.

HXH146E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH147E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH148E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH149E TBCREATE failed, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH150E TBADD failed, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH151E **No indexes are defined for *tablename***

Explanation: Self-explanatory. No indexes are available for accessing the indicated table. Access paths using an index are not valid for this SQL statement.

User Action: Specify a different access path and retry.

HXH152E **Too many selections (only 1 item may be processed at a time)**

Explanation: Only one index may be selected. You attempted to select more than one entry from the displayed list.

User Action: Select a single index and retry.

HXH153E **Invalid selection. The only valid selection characters are '/', 'S', and 'C'**

Explanation: An invalid selection character was entered.

User Action: Retry using one of the selection characters listed.

HXH161E **PREPARE failed, sqlcode *nnn***

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH162E **OPEN CURSOR failed, sqlcode *nnn***

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH163E ***tablename* is empty (contains no rows)**

Explanation: The specified plan table contains no data.

User Action: : Make sure that the correct user ID was entered. If the user ID is correct, then the plan table for that user contains no entries. You will need to populate the plan table (by binding with the Explain option) before using Express Hints.

HXH164E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH172E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH173E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH174E FETCH failed, sqlcode

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH175E Unable to locate SQL statement in DB2 catalog

Explanation: A processing error occurred while attempting to retrieve SQL statement information from the DB2 catalog. Although the statement was referenced by the plan table, it was not found in the catalog. Usually this is caused by obsolete information in the plan table.

User Action: Verify that the package referenced in the plan table (the one you are trying to create a hint for) really exists in the subsystem and that the SQL statements in the package have not changed since the plan table entry was created.

HXH176E An error occurred retrieving the SQL statement (premature end of data)

Explanation: A processing error occurred while retrieving SQL statement information from the DB2 catalog. The actual number of bytes retrieved did not match the predicted statement length. Usually this reflects the presence of corrupted data in SYSIBM.SYSPACKSTMT.

User Action: If possible, rebind the plan and retry. Otherwise contact HLS Technical Support.

HXH177E Unable to acquire SQLID, symptom=*nnn*

Explanation: An attempt to retrieve the current SQLID returned an unexpected condition code. In the message text, *nnn* is the sqlcode.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH178E Unable to set SQLID, symptom=*nnn*

Explanation: A SET CURRENT SQLID statement SQLID returned an unexpected condition code. In the message text, *nnn* is the sqlcode.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH179E Unable to set ophint, symptom=*nnn*

Explanation: A SET CURRENT OPTIMIZATION HINT statement SQLID returned an unexpected condition code. In the message text, *nnn* is the sqlcode.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH180E Unable to reset SQLID, symptom= *nnn*

Explanation: A SET CURRENT SQLID statement SQLID returned an unexpected condition code. In the message text, *nnn* is the sqlcode.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH181E COMMIT failed, sqlcode *nnn*

Explanation: Self explanatory. A COMMIT statement failed. In the message text, *nnn* is the sqlcode.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH183E Optimization hint rejected by DB2 -- *reason*

Explanation: DB2 could not validate the access path you specified (the access path is not valid for this SQL statement). In the message text, *reason* is a brief description of why the access path is not valid.

User Action: Specify a different access path and retry.

HXH188E COMMIT failed, sqlcode *nnn*

Explanation: Self explanatory. A COMMIT statement failed. In the message text, *nnn* is the sqlcode.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH189I Hint *hintname* has been added to *tablename* and is now available for use

Explanation: Self-explanatory. The hint has been created and is available for use.

User Action: None.

HXH193E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH194E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH195E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH197E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH198E TBCREATE failed, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH203E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH204E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH205E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.v

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH238E SQL error -- unable to determine status of Hints facility for subsystem *ssid*, sqlcode=*nnn*, sqlstate=*nnnnn*

Explanation: A failure has occurred in DB2, or in the Express Hints/DB2 interface. *nnn* represents an internal return code and is provided for diagnostic purposes.

User Action: Contact Technical Support

HXH240E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH258E TBADD failed for package entry, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH260E Invalid selection. The only valid selection characters are '/', 'S', and 'E'

Explanation: An invalid selection character was entered.

User Action: Retry using one of the selection characters listed.

HXH261E Invalid entry. Either select one item or else enter new sequence numbers for all.

Explanation: On the query elements screen, an invalid number of rows was selected, or incorrect action characters were entered.

User Action: Either select one single row for access path alterations, or else resequence the elements by entering all new sequence numbers.

HXH262E Invalid sequence numbers (*nnn* is missing)

Explanation: When changing the join sequence, the sequence numbers you enter must begin with one and must be contiguous, *i.e.*, no gaps (missing numbers) are permitted. You omitted sequence number *nnn*.

User Action: Retry using a valid range of sequence numbers.

HXH263E Invalid sequence numbers (*nnn* is duplicated)

Explanation: When changing the join sequence, the sequence numbers you enter must begin with one and must be contiguous, *i.e.*, no gaps (missing numbers) are permitted. You used sequence number *nnn* twice.

User Action: Retry using a valid range of sequence numbers.

HXH264E Zero is not a valid sequence number

Explanation: Join sequence numbers must begin with 1.

User Action: Retry using a valid range of sequence numbers.

HXH265E No hint name specified (you must assign a name)

Explanation: Self explanatory.

User Action: Use option 2 (NAME) on the Access Path Control submenu to assign a name to the new hint.

HXH267E Invalid input. Sequence numbers must be numeric.

Explanation: When changing the join sequence, the sequence numbers you enter must be numeric.

User Action: Retry using a valid range of sequence numbers.

HXH272E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH273E TBCREATE failed, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH274E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH275E TBADD failed for plan table, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

- HXH277E Too many selections (only 1 item may be processed at a time)**
- Explanation:** Only one entry may be selected on this panel. Multiple selection is not allowed.
- User Action:** Retry with a single selection.
-
- HXH280E TBCREATE failed, rc= *nnn***
- Explanation:** The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support
- User Action:** Contact Technical Support.
-
- HXH281E OPEN CURSOR failed, sqlcode *nnn***
- Explanation:** The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.
- User Action:** Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.
-
- HXH282E FETCH failed, sqlcode *nnn***
- Explanation:** The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.
- User Action:** Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.
-
- HXH283E Statement type not supported; complexity limit exceeded**
- Explanation:** The selected SQL statement is of a type or complexity not supported by this release of Express Hints. Statements containing subselects, for instance, are not presently supported.
- User Action:** None.
-
- HXH284E No hint name has been assigned (a name is required for a new hint)**
- Explanation:** Self explanatory.
- User Action:** Use option 2 (NAME) on the Access Path Control submenu to assign a name to the new hint.
-
- HXH285E Unable to process: optimization hints not enabled for subsystem *ssid***

Explanation: The optimization hints facility has not been activated for this subsystem. Express Hints requires the use of this facility. In the message text, *ssid* is the name of the subsystem.

User Action: Either (1) modify DSNZPARM to enable the optimization hints facility and recycle the subsystem, (2) if the subsystem is Version 8 or Version 9, run program HXH003 to enable optimization hints dynamically, or (3) use a different subsystem.

HXH290E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH295E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH296E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH297E Plan table retrieval error

Explanation: An unexpected error occurred when attempting to retrieve rows from a plan table.

User Action: Contact Technical Support.

HXH299E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

- HXH300E** **INSERT failed, sqlcode *nnn***
- Explanation:** The attempt to INSERT a row in a plan table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.
- User Action:** Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.
-
- HXH303E** **OPEN CURSOR failed, sqlcode *nnn***
- Explanation:** The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.
- User Action:** Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.
-
- HXH304E** ***tablename* contains no entries for this SQL statement; the statement has not been explained or is not explainable.**
- Explanation:** Self explanatory. The specified statement is not present in the indicated plan table. In the message text, *tablename* is the name of the target plan table.
- User Action:** None. This statement cannot be assigned a hint.
-
- HXH307E** **TBPUT failed , rc=*nnn***
- Explanation:** The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support
- User Action:** Contact Technical Support.
-
- HXH309E** **This access type does not use an index; index name should not be specified**
- Explanation:** You have specified an index name for a non-indexing access type.
- User Action:** Remove the index name and retry.
-
- HXH311E** **An index is required for this access type**
- Explanation:** You specified an access type that requires an index, but did not provide an index name.
- User Action:** Specify a valid index name and retry.
-
- HXH312E** **Syntax error: *indexname* is not a valid index name. Correct syntax is *owner.name*, e.g., 'JOE.INDEX_ONE'.**

Explanation: The index name you entered (*indexname*) has incorrect syntax.

User Action: Specify a valid index name and retry.

HXH315E Do not specify an index name for this type of access; indexes are chosen by DB2.

Explanation: Self explanatory.

User Action: Remove the index name and retry.

HXH801E Unable to connect to *ssss*, *rc=retcd*, *reas=rsn*

Explanation: Express Hints could not establish a connection to DB2. In the message text, *ssss* is the subsystem to which a connection was attempted, *retcd* is the return code from the connection request, and *rsn* is the reason code.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH802E Version *ver* of DB2 is not supported by this release of Express Hints.

Explanation: The requested subsystem is not running a release of DB2 that Express Hints does not support. In the message text, *ver* is the unsupported version.

User Action: None. This release of Express does not support the indicated version of DB2.

HXH803E Subsystem *ssss* is not available

Explanation: Express Hints was not able to establish a connection to subsystem *ssss*.

User Action: Verify that the correct subsystem name was entered on the Express Hints panel and that the subsystem is up and available. If this appears to be the case, contact HLS Technical Support.

HXH804E DSNTIAR failure, *rc=nnn*

Explanation: The DSNTIAR API issued an unexpected return code.

User Action: Contact HLS Technical Support.

HXH805E PREPARE failed, *sqlcode nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the *sqlcode* returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH806E *ssss* is a *vvv* subsystem. *vvv* is not supported.

Explanation: The requested subsystem is not running a release of DB2 that Express Hints does not support. In the message text, *ssss* is the subsystem ID and *vvv* is the version.

User Action: None. This release of Express does not support the indicated version of DB2.

- HXH807E Invalid selection. Enter one of the listed options.**
- Explanation:** The option entered is not valid on this panel.
- User Action:** Specify one of the valid options listed on the panel.
- HXH808E Limit exceeded (statement length) -- SQL statements longer than 32767 bytes are not supported**
- Explanation:** Self explanatory. The target SQL statement exceeds the supported length.
- User Action:** Select another SQL statement to process.
- HXH814E PREPARE failed, sqlcode *nnn***
- Explanation:** The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.
- User Action:** Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.
- HXH815E PREPARE failed, sqlcode *nnn***
- Explanation:** The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.
- User Action:** Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.
- HXH816E PREPARE failed, sqlcode *nnn***
- Explanation:** The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.
- User Action:** Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.
- HXH817E PREPARE failed, sqlcode *nnn***
- Explanation:** The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.
- User Action:** Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.
- HXH826E DISPLAY failed, rc=*nnn***
- Explanation:** The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support
- User Action:** Contact Technical Support.
- HXH827E DISPLAY failed, rc=*nnn***
- Explanation:** The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH828E **PREPARE failed, sqlcode** *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH829E **CREATE TABLE failed, rc=***nnn*

Explanation: The attempt to CREATE a table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH830E **COMMIT failed, rc=***nnn*

Explanation: A COMMIT SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH831I *ptname was created in dbname.tsname*

Explanation: Informational message. The plan table named *ptname* was successfully created in table space *tsname* in database *dbname*.

User Action: None.

HXH832E **PREPARE failed, sqlcode** *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH833E **CREATE TABLE failed, rc=***nnn*

Explanation: The attempt to CREATE a table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH834E **PREPARE failed, sqlcode** *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH835E **OPEN CURSOR failed, sqlcode** *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH836E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH837E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH838E DISPLAY failed, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH839E DISPLAY failed, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH840E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH841E CREATE TABLE failed, rc=*nnn*

Explanation: The attempt to CREATE a table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH842E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH843E CREATE TABLE failed, rc=nnn

Explanation: The attempt to CREATE a table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH844E PREPARE failed, sqlcode nnn

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH845E CREATE TABLE failed, rc=nnn

Explanation: The attempt to CREATE a table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH847E PREPARE failed, sqlcode nnn

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH848E CREATE TABLE failed, rc=nnn

Explanation: The attempt to CREATE a table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

User Action: None.

HXH849E PREPARE failed, sqlcode nnn

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH850E OPEN CURSOR failed, sqlcode nnn

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH851E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH852E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH853E INSERT failed, sqlcode *nnn*

Explanation: The attempt to INSERT a row in a plan table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH854E *ssid* does not allow the use of optimization hints

Explanation: Subsystem *ssid* is not configured to permit optimization hints to be used.

User Action: Modify the relevant DSNZPARM setting to enable optimization hints.

HXH855E Unable clear ophint, symptom=*nnn*

Explanation: Express Hints was unable to reset the OPTIMIZATION HINT special register. In the message text, *nnn* is a diagnostic code for use by Technical Support.

User Action: This condition should not occur in normal processing. Contact HLS Technical Support.

HXH856E EXPLAIN failed, rc=*nnn*

Explanation: An EXPLAIN ALL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: This condition should not occur in normal processing. Contact HLS Technical Support.

HXH857E Unable to capture SQLID, symptom=*nnn*

Explanation: Express Hints was unable to determine the user's current SQLID. In the message text, *nnn* is a diagnostic code for use by Technical Support.

User Action: This condition should not occur in normal processing. Contact HLS Technical Support.

HXH858E Unable to set SQLID, symptom=*nnn*

Explanation: Express Hints was unable to change the current SQLID. In the message text, *nnn* is a diagnostic code for use by Technical Support.

User Action: This condition should not occur in normal processing. Contact HLS Technical Support.

HXH859E Unable to reset SQLID, symptom=*nnn*

Explanation: Express Hints was unable to restore the user's SQLID. In the message text, *nnn* is a diagnostic code for use by Technical Support.

User Action: This condition should not occur in normal processing. Contact HLS Technical Support.

HXH860E COMMIT failed, rc=*nnn*

Explanation: A COMMIT SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH861E COMMIT failed, rc=*nnn*

Explanation: A COMMIT SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH862E Express Hints is not current in subsystem *ssss* ; please bind the Express Hints programs.

Explanation: Self explanatory. The current version of Express Hints has not been bound in subsystem *ssss*.

User Action: Run the BIND job supplied in the Express Hints SAMPLIB.

HXH863E DROP TABLE failed, rc=*nnn*

Explanation: The attempt to DROP a table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: This condition should not occur in normal processing. Contact HLS Technical Support.

HXH864E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH865E CREATE TABLE failed, rc=*nnn*

Explanation: The attempt to CREATE a table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH866E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH867E CREATE TABLE failed, rc=*nnn*

Explanation: The attempt to CREATE a table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH868E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH869E CREATE TABLE failed, rc=*nnn*

Explanation: The attempt to CREATE a table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH870E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH871E CREATE TABLE failed, rc=*nnn*

Explanation: The attempt to CREATE a table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH872E COMMIT failed, rc=*nnn*

Explanation: A COMMIT SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

- HXH873E PREPARE failed, sqlcode *nnn***
- Explanation:** The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.
- User Action:** Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.
- HXH874E DELETE failed, rc=*nnn***
- Explanation:** An attempt to delete a row from a work table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.
- User Action:** This condition should not occur in normal processing. Contact HLS Technical Support.
- HXH875E BIND QUERY failed for hint *hintname***
- Explanation:** A BIND QUERY command was not successful. This message is preceded by messages detailing the specific reason why BIND QUERY processing failed.
- User Action:** Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.
- HXH876E PREPARE failed, sqlcode *nnn***
- Explanation:** The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.
- User Action:** Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.
- HXH877E OPEN CURSOR failed, sqlcode *nnn***
- Explanation:** The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.
- User Action:** Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.
- HXH878E FETCH failed, sqlcode *nnn***
- Explanation:** The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.
- User Action:** Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.
- HXH879E Cost data is not available – check *stname***
- Explanation:** Express Hints cannot display cost information for an SQL statement; the user's DSN_STATEMNT_TABLE does not exist or cannot be accessed.
- User Action:** Make sure the user has a valid DSN_STATEMNT_TABLE defined in the correct subsystem.
- HXH880E PREPARE failed, sqlcode *nnn***

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, nnn is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH881E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, nnn is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH882E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, nnn is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH883E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, nnn is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH884E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, nnn is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH885E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, nnn is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH886E CREATE LOB TABLESPACE failed, sqlcode *nnn*

Explanation: The attempt to create a table space for an auxiliary table was not successful. In the message text, nnn is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH900E Express Hints experienced a problem during shutdown, symptom=*nnn*

Explanation: An unexpected error was detected during termination processing. This message should not appear in normal processing.

User Action: Contact Technical Support.

HXH901E **Unable to connect to *ssid*, rc=*cond*, reas=*nnn***

Explanation: The attempt to connect to DB2 failed for an unexpected reason. In the message text, *ssid* is the subsystem name, *cond* is the condition code returned from Call Attach, and *nnn* is the reason code.

User Action: Use the DB2 manuals to determine the reason for the Call Attach failure and correct the underlying condition.

HXH902E **Version *xxx* of DB2 is not supported by this release of Express Hints**

Explanation: Self explanatory. In the message text, *xxx* is the DB2 version of the subsystem.

User Action: None.

HXH903E **Subsystem *ssid* is not available**

Explanation: The specified subsystem does not exist, is not up, or is not accepting connect requests. In the message text, *ssid* is the subsystem name. This error is often caused by spelling the subsystem name incorrectly.

User Action: Make sure the subsystem name is entered correctly and that the subsystem is up.

HXH904E **DSNTIAR failure, rc=*nnn***

Explanation: The DSNTIAR diagnostic routine terminated with an unexpected condition code. In the message text, *nnn* is the return code from DSNTIAR.

User Action: Determine why DSNTIAR is unavailable or unusable. Correct the problem and retry.

HXH905E **PREPARE failed, sqlcode *nnn***

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH906I ***ssid* is a *vvv* subsystem. *vvv* is not supported.**

Explanation: The target subsystem is running a version of DB2 not supported by Express Hints. In the message text, *ssid* is the subsystem name and *vvv* is the (unsupported) version of DB2 running there.

User Action: None.

HXH907E Invalid selection. Enter one of the listed options.

Explanation: The user entered an option not recognized by the program.

User Action: Choose one of the options listed on the panel.

HXH908E Limit exceeded (statement length) -- SQL statements longer than 32767 bytes are not supported

Explanation: Self explanatory.

User Action: None.

HXH909E Unicode initialization error rc=*cond* reason=*rsn*

Explanation: The conversion API did not initialize successfully; Unicode System Services returned an unexpected condition code. In the message text, *cond* is the return code and *rsn* is the associated diagnostic reason code.

User Action: Refer to the Unicode System Services documentation to determine why conversion could not be performed and correct the underlying reason.

HXH910E Unicode conversion error rc=*cond* reason=*rsn*

Explanation: The SQL text could not be converted from Unicode to EBCDIC. In the message text, *cond* is the condition code issued by Unicode System Services and *rsn* is the associated diagnostic reason code.

User Action: Refer to the Unicode System Services documentation to determine why conversion could not be performed and correct the underlying reason.

HXH911E Unicode conversion error rc=*cond* reason=*rsn*

Explanation: The SQL text could not be converted from Unicode to EBCDIC. In the message text, *cond* is the condition code issued by Unicode System Services and *rsn* is the associated diagnostic reason code.

User Action: Refer to the Unicode System Services documentation to determine why conversion could not be performed and correct the underlying reason.

HXH914E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH915E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH916E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH917E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH918E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH919E TBADD failed for package entry, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH920E Unicode conversion error rc=*cond* reason=*rsn*

Explanation: The SQL text could not be converted from Unicode to EBCDIC. In the message text, *cond* is the condition code issued by Unicode System Services and *rsn* is the associated diagnostic reason code.

User Action: Refer to the Unicode System Services documentation to determine why conversion could not be performed and correct the underlying reason.

HXH921E TBCREATE failed, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH922E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH923E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH924E No current SQL statements for this package were found in the catalog

Explanation: The package was found in the catalog but no SQL statements were present. This condition should not occur in normal operation; the most likely cause is corrupted data in the catalog or in the DBRM that was bound to create the package.

User Action: Contact Technical Support.

HXH3001E Express Hints failed during initialization, symptom=*nnn*

Explanation: A failure has occurred in ISPF, or in the Express Hints/ISPF interface. *nnn* is an internal return code and is provided for diagnostic purposes.

User Action: This condition should not occur in normal processing. Contact HLS Technical Support.

HXH3003E Unable to connect to subsystem *ssss*

Explanation: Express Hints was not able to establish a connection to subsystem *ssss*.

User Action: Verify that the correct subsystem name was entered on the panel and that the subsystem is up and available. If this appears to be the case, contact HLS Technical Support.

HXH3009E TBCREATE failed, rc= *nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH3019E Too many selections (only 1 item may be processed at a time)

Explanation: Only one package may be processed at a time. You attempted to select more than one entry from the displayed list.

User Action: Select a single entry and retry.

HXH3020E Invalid selection - the only valid selection characters are 'I', 'S', and 'K'

Explanation: An invalid selection character was entered.

User Action: Retry using one of the selection characters listed.

HXH3051E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3070E *ptname* is not usable -- Express Hints requires

Explanation: PLAN_TABLEs can be created in various formats with different numbers of columns. In order to use Express Hints, your PLAN_TABLE must have 58 columns or more; PLAN_TABLEs with fewer than 58 columns lack some of the data needed for Express Hints processing. In the message text, *ptname* is the name of the unusable PLAN_TABLE.

User Action: Recreate the PLAN_TABLE in the 58-column (or later) format.

HXH3077E A hint name must be specified

Explanation: Each optimization hint must be named.

User Action: Enter a valid 1 to 8 character hint name.

HXH3082E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3083E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3084E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3093E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3094E INSERT failed, sqlcode *nnn*

Explanation: The attempt to INSERT a row in a plan table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3110E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3146E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3147E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3148E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3149E TBCREATE failed, rc= *nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH3150E TBADD failed, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH3151E No indexes are defined for *tablename*

Explanation: Self-explanatory. No indexes are available for the indicated table. Access paths requiring an index are not valid for this SQL statement.

User Action: Specify a different access path.

HXH3152E Too many selections (only 1 item may be processed at a time)

Explanation: Only one index may be specified. You attempted to select more than one entry from the displayed list.

User Action: Select a single index.

HXH3153E Invalid selection - the only valid selection characters are '/', 'S', and 'C'

Explanation: An invalid selection character was entered.

User Action: Retry using one of the selection characters listed.

HXH3154E 'xxx' is not a valid command

Explanation: DEBUG is the only command that is valid on this panel. No other command is acceptable.

User Action: Enter a valid command or else leave the command field blank.

HXH3161E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3162E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3163E *ptname is empty (contains no rows)*

Explanation: The user's plan table contains no rows.

User Action: This condition should not occur in normal processing. Contact HLS Technical Support.

HXH3164E **FETCH failed, sqlcode** *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, nnn is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3172E **PREPARE failed, sqlcode** *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, nnn is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3173E **OPEN CURSOR failed, sqlcode** *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, nnn is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3174E **FETCH failed, sqlcode** *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, nnn is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3177E **Unable to capture SQLID, symptom=***nnn*

Explanation: Express Hints was unable to determine the user's current SQLID. In the message text, nnn is a diagnostic code for use by Technical Support.

User Action: This condition should not occur in normal processing. Contact HLS Technical Support.

HXH3178E **Unable to set SQLID, symptom=***nnn*

Explanation: Express Hints was unable to change the current SQLID. In the message text, nnn is a diagnostic code for use by Technical Support.

User Action: This condition should not occur in normal processing. Contact HLS Technical Support.

HXH3179E Unable set ophint, symptom=*nnn*

Explanation: Express Hints was unable to set the OPTIMIZATION HINT special register. In the message text, *nnn* is a diagnostic code for use by Technical Support.

User Action: This condition should not occur in normal processing. Contact HLS Technical Support.

HXH3180E Unable to reset SQLID, symptom=*nnn*

Explanation: Express Hints was unable to restore the user's SQLID. In the message text, *nnn* is a diagnostic code for use by Technical Support.

User Action: This condition should not occur in normal processing. Contact HLS Technical Support.

HXH3183E Optimization hint rejected by DB2 -- *reason*

Explanation: The specified access path is not acceptable to DB2. In the message text, *reason* is an explanation of the reason the access path was rejected.

User Action: Specify an acceptable access path and retry.

HXH3188E COMMIT failed, rc=*nnn*

Explanation: A COMMIT SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3189I Hint *nnnn* has been loaded into the repository and is now available for use

Explanation: Self explanatory. In the message text, *nnnn* is the hint name.

User Action: None.

HXH3193E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3194E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3195E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3197E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3198E TBCREATE failed, rc= *nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH3199E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3200E TBADD failed, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH3201E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3238E SQL error -- unable to determine status of Hints facility for subsystem *ssss*, sqlcode=*xxxx*, sqlstate=*yyyy*

Explanation: Express Hints encountered an unexpected condition when trying to determine whether optimization hints is enabled for the indicated subsystem. In the message text, *ssss* is the subsystem ID and *xxxx* and *yyyy* are the sqlcode and sqlstate, respectively, for the failing SQL statement.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3258E COMMIT failed, rc=*nnn*

Explanation: A COMMIT SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3260E Invalid selection - the only valid selection characters are '/', 'S', and 'E'

Explanation: An invalid selection character was entered.

User Action: Retry using one of the selection characters listed.

HXH3261E Invalid entry. Either select one item or else enter new sequence numbers for all.

Explanation: An invalid number of rows was selected, or incorrect action characters were entered.

User Action: Either select one single row for access path alterations, or else resequence the elements by entering all new sequence numbers.

HXH3262E Invalid sequence numbers ('nnn' is missing)

Explanation: Self explanatory. New sequence numbers were specified but they are not complete. The indicated sequence number is not present

User Action: Enter a valid range of sequence numbers.

HXH3263E Invalid sequence numbers ('nnn' is duplicated)

Explanation: Self explanatory. New sequence numbers were specified but they are not complete. The indicated sequence number is not present

User Action: Enter a valid range of sequence numbers.

HXH3264E Zero is not a valid sequence number

Explanation: Self explanatory. A sequence number of 0 was entered on some line. Zero is not a valid query element sequence number.

User Action: Enter a valid range of sequence numbers.

HXH3265E No hint name specified (you must assign a name)

Explanation: Each optimization hint must be named.

User Action: Enter a valid 1 to 8 character hint name.

HXH3267E Invalid input. Sequence numbers must be numeric.

Explanation: Self explanatory. When reordering query elements, the sequence numbers you specify must be all numeric.

User Action: Correct the input and retry.

HXH3279E PREPARE failed, sqlcode nnn

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, nnn is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3280E TBCREATE failed, rc= *nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH3281E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3282E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3283E Statement type not supported; complexity limit exceeded.

Explanation: The selected SQL statement is of a type or complexity not supported by this release of Express Hints. Statements containing subselects, for instance, are not presently supported.

User Action: Correct the input and retry.

HXH3284E No hint name has been assigned (a name is required for a new hint)

Explanation: Each optimization hint must be named.

User Action: Enter a valid 1 to 8 character hint name.

HXH3285E Unable to process: optimization hints not enabled for subsystem *ssid*

Explanation: Subsystem *ssid* is not configured to permit optimization hints to be used.

User Action: Modify the relevant DSNZPARM setting to enable optimization hints.

HXH3290E FETCH failed, sqlcode *nnn*

Explanation: The attempt to FETCH a successfully opened cursor failed. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3295E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3296E OPEN CURSOR failed, sqlcode *nnn*

Explanation: The attempt to OPEN a cursor was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3297E Plan table retrieval error

Explanation: PLAN_TABLE did not contain an expected row.

User Action: This condition should not occur in normal processing. Contact HLS Technical Support.

HXH3299E PREPARE failed, sqlcode *nnn*

Explanation: The attempt to PREPARE a dynamic SQL statement was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3300E INSERT failed, sqlcode *nnn*

Explanation: The attempt to INSERT a row in a plan table was unsuccessful. In the message text, *nnn* is the sqlcode returned by DB2.

User Action: Use the DB2 messages and codes manuals to determine the reason for the failure and correct the underlying condition.

HXH3307E TBPUR failed, rc=*nnn*

Explanation: The ISPF API issued an unexpected return code. This should not occur in normal operation. In the message text, *nnn* is a diagnostic code for use by Technical Support

User Action: Contact Technical Support.

HXH3309E This access type does not use an index; index name should not be specified

Explanation: The access type you have specified does not use indexing.

User Action: Do not specify an index name.

HXH3311E An index is required for this access type

Explanation: You specified an access type that uses indexing.

User Action: Enter a valid index name.

HXH3312E Syntax error: *nnnn* is not a valid index name. Correct syntax is owner.name, e.g., 'JOE.INDEX_ONE'.

Explanation: The specified index name has incorrect syntax. In the message text, *nnnn* is the invalid index name.

User Action: Enter a valid index name.

HXH3315E Do not specify an index name for this type of access; indexes are chosen by DB2.

Explanation: You cannot select the index for the specified access path. The DB2 optimizer will choose an appropriate index.

User Action: Omit the index name.

Glossary

A

access path

An access path is a representation of the execution control data that is used to execute an SQL Statement.

access type

There are varied access types that determine how an SQL Statement will obtain data.

authid

This is the authorization Id of whomever is trying to execute an SQL Statement.

B

bind

The process by which SQL statements are finalized for Static SQL.

C

CLIST

Command List is a series of instructions that are executed interactively with TSO.

D

DB2 optimizer

The DB2 Optimizer reads SQL statements at preprocessor time for a Static SQL or at load time for a Dynamic SQL. Based on the structure of the SQL Statement and related data about the tables being processed, the optimizer decides the access path to use for the execution of the SQL Statement.

DBRM

Data Base Resource Module is a collection of SQL data that is used to represent the SQL for a package. It is a separate entity that can be moved between subsystems and then bound.

DSN_STATEMENT_CACHE_TABLE

This table is updated by writing cache information from an active DB2 subsystem. It contains the SQL text and related data. However, it does not contain the same information found in the Plan Table.

DSN_STATEMENT_HISTORY_TABLE

A table that contains accumulated history from repeated executions of HXH100 to capture the data from the DSN_STATEMENT_CACHE_TABLE.

DSN_STATEMENT_TABLE

This table contains estimates of execution cost. The estimates are sometimes from the actual execution costs. We have found that within a Version of DB2 they are consistent with each other. That is to say if the estimate times double then the actual execution times will approximate the same doubling.

DSN_USERQUERY_TABLE

This table is used for the control of hint processing.

dynamic SQL

Dynamic SQL is SQL that is reexamined for selection of its access path each time the SQL is loaded in DB2 or each time it is loaded into the DB2's Cache.

E

explain

EXPLAIN is either the processing of an SQL Statement to identify the access path information used to execute an SQL Statement or the reading and presentation of that data.

I

index

An index is a DB2 table associated with another DB2 table. It contains an ordered limited set of the data in the primary table. It is used to point to within the table to speed processing.

ISPF

ISPF is a TSO command that allows easier user interface through TSO.

J

JCL

JCL is Job Control Language, which states the requirements for execution of a batch program to the Operating System.

O

optimization hints

Optimization Hints is the process that allows you to set an access path hint for a given SQL Statement. The hint is processed when the SQL Statement is loaded for execution. The return code from the execution states whether or not the hint was accepted.

P

package

A package is a collection of SQL Statements that relate to a program.

plan table

Either refers to a single PLAN Table or the collection of tables that are created when doing an EXPLAIN. The data in the PLAN TABLE consists of access path information for an SQL Statement. The information may be a single row or multiple rows. Most organizations have an EXPLAIN program that takes the information within the PLAN TABLE and represents it in a more readable format.

prefetch

DB2's attempt to bring in data before it is passed to the program for processing.

S

SQL statement

Structure Query Language. An SQL Statement is a single statement that is presented for preprocessing.

static SQL

Static SQL is SQL that is preprocessed and then bound. The access path information about the SQL is not changed unless the SQL is rebound.

T

tables

Tables are physical data entities that hold data for DB2.

TSO

Time Share Option.

Index

A

- access path dynamic SQL
 - modifying, 16
 - specifying, 15
 - viewing, 16
- access path static SQL
 - modifying, 10
 - specifying, 9
- activating
 - Express Hints, 7

B

- basic parameters dynamic SQL
 - specifying, 14
- basic parameters static SQL
 - specifying, 8
- batch reports
 - creating, 24
 - descriptions, 26
 - using, 24

C

- comparing
 - cost, 19
- complete list of key index columns
 - showing, 21
- cost
 - comparing, 19

creating

- batch reports, 24
- hint for dynamic SQL, 18
- hint for static SQL, 12
- statement cost compare, 19

D

- displaying
 - index columns, 21
 - package summary, 23
- dynamic SQL
 - using, 14

E

- Express Hints
 - activating, 7
 - installing, 4
 - what, 2
 - why, 2

G

- glossary, 31

H

- hint for dynamic SQL
 - creating, 18
- hint for static SQL
 - creating, 12

I

- index columns
 - showing, 21
- installing
 - Express Hints, 4
 - planning, 4
 - running the jobs, 6
- introduction, 2

M

- modifying
 - access path dynamic SQL, 16
 - access path static SQL, 10

O

- other features
 - using, 21

P

- package
 - selecting, 9, 15
- package summary
 - showing, 23
- planning
 - install, 4
- product files
 - unpacking, 5
 - uploading, 5

R

- running

- installation jobs, 6
- setup CLIST, 5

S

- samples
 - batch reports, 26
- security
 - setting, 7
- selecting
 - package, 9, 15
 - SQL statement, 9, 15
- setting
 - security, 7
- setup CLIST
 - running, 5
- showing
 - complete list of key index columns, 21
 - package summary information, 23
- specifying
 - access path dynamic SQL, 15
 - access path static SQL, 9
 - basic parameters dynamic SQL, 14
 - basic parameters static SQL, 8
- SQL statment
 - selecting, 9, 15
- statement cost compare
 - creating, 19
- static SQL
 - using, 8

U

- unpacking

- product files, 5
- uploading
 - product files, 5
- using
 - batch reports, 24
 - dynamic SQL, 14

- other features, 21
- static SQL, 8

V

- viewing
 - access path dynamic SQL, 16

Notices

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

HLS TECHNOLOGIES, INC. PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR 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. HLS may make improvements and/or changes in the product(s) and or the program(s) described in this publication at any time without notice.

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

Trademarks

IBM, DB2, MVS, ISPF, OS/390 and z/OS are trademarks of IBM Corporation.