Pivotal™

PRODUCT DOCUMENTATION

# Pivotal ADS

Version 1.1.4

## Release Notes

Rev: A02

# Table of Contents

1 Welcome to Pivotal Advanced Database Services 1.1.4 ____ 4
1.1 About PADS Components ____ 4
1.1.1 HAWQ ____ 4
1.1.2 PXF ____ 4
1.1.3 MADlib ____ 5
2 New Features ____ 6
2.1 User-Defined Functions ____ 6
2.1.1 UDF support in PADS 1.1.4: ____ 6
2.1.2 UDF features not supported in PADS 1.1.4: ____ 6
2.2 HAWQ toolkit ____ 6
3 About the ADS 1.1.4 Release ____ 7
3.1 Supported Platforms ____ 7
3.2 Installation options ____ 7
4 Resolved Issues in PADS 1.1.4 ____ 8
5 Known Issues in PADS 1.1.x ____ 10
5.1 Known Issues in HAWQ 1.1.4.0 ____ 10
5.2 Known Issues in HAWQ 1.1.3.0 ____ 11
5.3 Known Issues in HAWQ 1.1.0.3 ____ 12
5.4 Known Issues in HAWQ 1.1.0.1 ____ 12
5.5 Known Issues in PXF 2.x.x ____ 13
6 Pivotal and PADS Interoperability ____ 15
7 PADS 1.1.4 and Pivotal HD Documentation ____ 16
8 Upgrading to HAWQ 1.1.x ____ 17
8.1 Upgrading from HAWQ 1.1.x to HAWQ 1.1.y ____ 17
8.2 Upgrading from HAWQ 1.0.x to HAWQ 1.1.x ____ 18
8.3 Preupgrade Preparation ____ 18
8.4 Install the HAWQ Software Binaries ____ 18
9 Upgrade Execution ____ 19
9.1 Post Upgrade on your HAWQ 1.1.x System ____ 19
10 Troubleshooting a Failed Upgrade ____ 22

GoPivotal, Inc. believes the information in this publication is accurate as of its publication date. The information is subject to change without notice. THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." GOPIVOTAL, INC. ("Pivotal") MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

**Use of Open Source**

This product may be distributed with open source code, licensed to you in accordance with the applicable open source license. If you would like a copy of any such source code, Pivotal will provide a copy of the source code that is required to be made available in accordance with the applicable open source license. Pivotal may charge reasonable shipping and handling charges for such distribution.

# 1 Welcome to Pivotal Advanced Database Services 1.1.4

Pivotal Advanced Database Services (ADS), extends Pivotal Hadoop (HD) Enterprise, adding rich, proven parallel SQL processing facilities. These SQL processing facilities enhance productivity, rendering Hadoop queries faster than any Hadoop-based query interface on the market. Pivotal ADS enables data analysis for a variety of Hadoop-based data formats using the Pivotal Extension Framework (PXF), without duplicating or converting HBase files. For the best performance, you can use an optimized format for Pivotal ADS table storage.

## 1.1 About PADS Components

PADS comprises the following components:

- HAWQ
- PXF
- MADlib

### 1.1.1 HAWQ

HAWQ is a parallel SQL query engine that combines the key technological advantages of the industry-leading Greenplum Database with the scalability and convenience of Hadoop. HAWQ reads data from and writes data to HDFS natively.

Using HAWQ functionality, you can interact with petabyte range data sets. HAWQ provides users with a complete, standards compliant SQL interface.

Leveraging Greenplum Database’s parallel database technology, HAWQ consistently performs tens to hundreds of times faster than all Hadoop query engines in the market.

### 1.1.2 PXF

PXF enables SQL querying on data in the Hadoop components such as HBase, Hive, and any other distributed data file types. These queries execute in a single, zero materialization and fully-parallel workflow. PXF also uses the PADS advanced query optimizer and executor to run analytics on these external data sources, or transfers it to PADS to analyze locally. PXF connects Hadoop-based components to facilitate data joins, such as between HAWQ tables and HBase table. Additionally, the framework is designed for extensibility, so that user-defined connectors can provide parallel access to other data storage mechanisms and file types.

#### PXF Interoperability

PXF operates as an integral part of PADS, and as a light add-on to Pivotal HD. On the database side, PXF leverages the external table custom protocol system. Therefore, creating a PXF table provides the same interoperability as an external table in Greenplum Database. On the Pivotal HD side, the PXF component physically lives on the Namenode and each or some Datanodes. It operates mostly as a separate service and does not interfere with Hadoop components internals.

## 1.1.3 MADlib

MADlib is an open-source library for scalable in-database analytics. It provides data-parallel implementations of mathematical, statistical and machine learning methods for structured and unstructured data. MADlib combines the efforts used in commercial practice, academic research, and open-source development. You can find more information at http://madlib.net.

# 2 New Features

This section describes the new features specific to PADS 1.1.4

For specific information about a previous release, please refer to the associated release notes.

## 2.1 User-Defined Functions

PADS 1.1.3 provided support for User-Defined Functions (UDF). The UDF feature extends the functionality of the HAWQ database by providing functions that can be evaluated in SQL statements. With each release, Pivotal extends UDF feature support. This section lists the UDF functionality supported in each release, and explicitly calls out features that are not supported.

### 2.1.1 UDF support in PADS 1.1.4:

- User defined aggregates
- Procedural languages:
  - PL/Python
  - PL/R

### 2.1.2 UDF features not supported in PADS 1.1.4:

- SECURITY DEFINER when creating functions
- Set returning functions
- TABLE functions
- Nested functions error out during execution
- User-Defined base type
- SORTOP
- Window functions
- ALTER set encoding, set schema, rename

See the *Pivotal HAWQ Installation Guide* for information about requirements and installation.

See the *Pivotal HAWQ Administrator Guide* for detailed information about features and usage.

## 2.2 HAWQ toolkit

The hawq_toolkit is an administrative schema that users can use to query system catalogs, log files, and operating environment for system status information. See the *Pivotal HAWQ Administrator Guide* for detailed information.

# 3 About the ADS 1.1.4 Release

This section lists the supported platforms and installation options for this release

## 3.1 Supported Platforms

PADS 1.1.4 supports the following platforms:

- Red Hat Enterprise 6.4-64 bit and 6.2-64 bit
- CentOS 6.4-64 bit and 6.2-64 bit

## 3.2 Installation options

There are two ways to install HAWQ.

- Stand alone install – Please see *HAWQ 1.1 Installation Guide*
- Pivotal Command Center Command Line Interface– Please see *Pivotal HD Enterprise 1.0 Installation and Administrator Guide.*

# 4 Resolved Issues in PADS 1.1.4

The table below lists issues that are now resolved in PADS 1.1.4.

For issues resolved in prior releases, refer to the corresponding release notes available from Support Zone

**Table: Resolved Issues in PADS 1.1.4**

| Issue | Category | Resolved in | Description |
|---|---|---|---|
| ARD-132 | PXF | PADS 1.1.3 | During a join between two PXF tables, the optimizer may periodically hold the HBase side of the scan. In such a situation, the PXF HBase scanner timeout exception is not caught.<br>**Workaround** : Increase the HBase scanner timeout. |
| HAWQ-1032 | AO tables and Column Store | PADS 1.1.3 | If you inserted a large block of data, for example about 2M, in a single query, HAWQInputFormat will not be able to access it.<br>**Workaround** : insert the data in multiple querieS. |
| HAWQ-1031 | AO tables and Column Store | PADS 1.1.3 | You will see the NullPointerException if you try to access a column using the wrong column name. |
| HAWQ-1023 | HDFS Access LAyer | PADS 1.1.3 | HAWQInputFormat throws the ClassCastException if it reads an invalid metadata file. |
| HAWQ- 990 | AO tables and Column Store | PADS 1.1.3 | AOInputFormat: bytesTo DecimalStr throws the ArrayIndexOutOf BoundsException. Occurs when a table has columns with datatypes that are greater and less than 8 characters long. For example, the table may contain timestamps and int4. If all the 8-length columns in a tuple have a null value, HAWQInputFormat throws the ArrayIndexOutOfBoundsException exception. |
| HD-2362 | PXF | PADS 1.1.1 | The select query fails for an external table created on views for any HIVE table. A more comprehensive error message is now given. |
| HAWQ-282 | PXF | PADS 1.1.1 | The ANALYZE command silently ignored an incorrect Analyzer name when analyzing a PXF table. A WARNING is now issued. |
| HAWQ-246 | PXF | PADS 1.1.1 | A segmentdb could not read a block located on the same machine. The algorithm was enhanced to account for cases where PXF services are located on a remote region server. |

| Issue | Category | Resolved in | Description |
|---|---|---|---|
| HAWQ-245 | PXF | PADS 1.1.1 | Could not add PXF-specific logic to gp_external_max_segments. Modifying this parameters allows you to achieve the maximum possible distribution on hosts. You can also limit the number of segments that are running. |
| HAWQ-216 | Management Tools | PADS1.1 | gpstart failed if the cluster is busy. This issue did not happen frequently, since the system is idle at start time. However, it is important to remember that it could happen because the system may have other non-HAWQ workload on the same cluster. |

# 5 Known Issues in PADS 1.1.x

This section lists all the known issues in PADS 1.1.x. You will find issues and descriptions categorized under their respective components. A workaround is provided where applicable. This section contains the following topics:

- Known Issues in HAWQ 1.1.4.0
- Known Issues in HAWQ 1.1.3.0
- Known Issues in HAWQ 1.1.0.0
- Known Issue in HAWQ 1.1.0.1
- Known Issues in PXF 2.x.x

**Important**: Pivotal Hadoop (PHD) 1.1. has a new High Availability (HA) feature. This feature has the following known issue with HAWQ: If NameNode HA is configured in a cluster, HAWQ is unable to take advantage of HA capability when the primary configure HA for HDFS. In the case of the primary NameNode failing, HAWQ will not be able to failover to the secondary NameNode. Manual re-direction to secondary NameNode is not currently supported in this release. Therefore we highly recommend that HA should be disabled when used with HAWQ.

## 5.1 Known Issues in HAWQ 1.1.4.0

| Issue | Category | Description |
|---|---|---|
| HAWQ-1610 | Build and Installer | PL/R package changes.<br>Check the name of your plr package. If it is plr-1.1.4.0-5152.x86_64.tgz,download the latest version plr-1.1.4.0-5664.x86_64.tgz for HAWQ 1.1.4.0 from Pivotal. The new package contains the file plr.sql with the necessary PL/R helper functions. |
| HAWQ-1418 | Catalog and Metadata | HAWQ 1.1.4.0 does not support aggregate derived functions. |

| Issue | Category | Description |
|---|---|---|
| HAWQ-1260 | Query Execution | A certain class of uncorrelated subqueries are known to fail. The subquery should have a user defined object and a distributed table. For example:<br>SELECT * FROM t1 WHERE t1.a < (SELECT foo(t2.b) FROM t2 LIMIT 1);<br>In this example, the subquery "SELECT foo(t2.b) FROM t2 LIMIT 1" has no correlation with the outer query. The subquery also invokes the UDF "foo()" and queries a distributed table "t2". Another example:<br>SELECT array(SELECT foo(t1.a) FROM t1);<br>Such type of queries fail with the following error:<br>ERROR cache lookedup failed for ... |
| HAWQ-1379 | Management Tool | hawq_toolkit cannot be used directly after upgrading from an old version. This is because toolkit related objects are not created in the old version.<br>**Workaround**: for each existing database instance where a user wants to use hawq_toolkit, perform following steps as superuser:<br>1. create a new schema named hawq_toolkit: psql -q -c "CREATE SCHEMA hawq_toolkit" $DATABASE_NAME<br>2. create toolkit related objects: psql -q -f $INSTALL_DIR/share/postgresql/gp_toolkit.sql $DATABASE_NAME<br>After performing the above operations on template1, every newly created database using template1 as template database, will have hawq_toolkit automatically, meaning no need to perform the above operation. |
| HAWQ-1368 | Management Tool | The view, hawq_size_of_database, does not check user permission of those databases and only reports sizes of all user databases. |
| HAWQ-1270 | Management Tool | The user must have access permission to the view, hawq_size_of_schema_disk. |
| HAWQ-1369 | Management Tool | When the underlying HDFS is online, hawq_size_of_database includes the data size on both HDFS and local storage of the master; when the HDFS is offline, that view only has the data size on local storage of the master. |

## 5.2 Known Issues in HAWQ 1.1.3.0

| Issue | Category | Description |
|---|---|---|
| HAWQ-1167 | Performance | Enabling Kerberos shows a 10% downgrade in HAWQ performance. |

| Issue | Category | Description |
|---|---|---|
| HAWQ-1099 | Connectivity | If you enable kerberos authentication, the ODBC function SQL GetInfo returns an incorrect version of HAWQ. |
| HAWQ-1078 | Query Execution | Continuously issue deepslice queries cause error in HDFS with kerberos. |
| HAWQ-1056 | DML | Inserting data into a temp table generates an Append-only Storage Write error. |
| HAWQ-1022 | Utility commands | Default value for dfs.client.socket-timeout in HDFS setting in gpcheck need to be corrected. |

## 5.3 Known Issues in HAWQ 1.1.0.3

| Issue | Category | Description |
|---|---|---|
| HAWQ-859 | Query Optimizer | pg_dumpall test suite runs slowly<br>The overhead is due to the command pg_dumpall. pg_dumpall generates multiple queries over the catalog tables. Since ORCA optimizes these queries. Although these are simple queries, ORCA adds the overhead.<br>**Workaround** : Turn ORCA off. |

## 5.4 Known Issues in HAWQ 1.1.0.1

| Issue | Category | Description |
|---|---|---|
| HAWQ-256 | Storage | HAWQ does not support kerberos authentication on HDFS. |
| HAWQ-255 | Network | HAWQ does not support the IPv6 protocol. |
| HAWQ-225 | Storage | When the number of partitions or columns of a column oriented table is large or write concurrency is high, HAWQ encounters an HDFS concurrency write limitation. Data loading performance may degrade and fail.<br>**Workaround:** for partitioned tables, load data partitions one by one, instead of loading all the data randomly to all the partitions. |
| HAWQ-224 | Backup and Restore | Only non-parallel logical backup and restore is supported. Pivotal recommends that you use physical backup and restore. |

| Issue | Category | Description |
|---|---|---|
| HAWQ-26 | DDL | duplicate key violates unique constraint pg_type_typname_nsp_indexWhen two sessions attempt to create a table with the same name and in the same namespace, one of the sessions will error out with a less user-friendly error message of the form "duplicate key violates unique constraint". |

## 5.5 Known Issues in PXF 2.x.x

| Issue | Category | Description |
|---|---|---|
| HAWQ-1394 | PXF 2.1.1 | When using PXF to communicate with a kerberized Pivotal Hadoop, PXF assumes that P-HD is using port 8020. If that is not the case, PXF will fail to communicate and transfer data. You will see the following message:<br>ERROR: fail to get filesystem credential for uri hdfs://<namenode>:8020/ (cdbfilesystemcredential.c:194) |
| HAWQ-1380 | PXF 2.1.1 | Query conditions specified with more than one element are not pushed down to PXF.<br>No workaround is necessary since the query is filtered by HAWQ and returns the correct results. This behavior only impacts the performance of HBase and Hive. |
| HAWQ-1142 | PXF 2.0.3 | If you write the ANALYZE command on PXF writable tables it fails with an error.<br>**Workaround** : Add FRAGMENTER or PROFILE parameters to the location of the PXF writable table. The ANALYZE command will issue a warning, but will not fail. |
| HAWQ-1133 | PXF 2.0.3 | A query might fail if the same DN has to read several BZip2 jobs at the same time. This is caused because BZip2 codec is not thread safe.<br>**Workaround** : Pivotal recommends that you do not read more than one BZip2 files at the same time. |
| HAWQ-1111 | PXF 2.0.3 | HAWQ does not report the details of a BadRecordException error message generated in PXF.<br>**Workaround** : You must look for the details in the PXF log on HDFS. |
| HAWQ-1141 | PXF 2.0.2 | If the rows in a Hive array are in different sizes, HiveResolver returns incorrect data. |
| HD-6122 | PXF 2.0.2 | PXF only works with HiveServer1, not HiveServer2. |

| Issue | Category | Description |
|---|---|---|
| HAWQ-291 | PXF 2.0.1 | HDFS does not work properly when accessing data files that contain header rows.<br>**Workaround** : Specify an error table with the "no HEADER" flag. |

# 6 Pivotal and PADS Interoperability

Pivotal releases a number of client tool packages on various platforms that can be used to connect to PADS. The following table describes the client tool package compatibility with PADS. Client tool packages are available at the EMC Download Center.

**Table: Interoperability Matrix**

| Client package | Description | Operating system | Client version | HAWQ version |
|---|---|---|---|---|
| Connectivity | Standard PostgreSQL Database Drivers (ODBC, JDBC) | Windows 2008 RedHat 6.4 and 6.2, 64 bit | 4.2.6SP | 1.1.4 |
| HAWQ Client | Command Line Interface | Windows 2008 RedHat 6.4 and 6.2, 64 bit | 4.2.6SP | 1.1.4 |
| Pivotal Command Center | A web-based tool for managing and monitoring your Pivotal HD cluster.<br>Note: Pivotal Command Center 2.0.x does not support DCA V1, DCA V2 or Greenplum Database. | Windows 2008 RedHat 6.4 and 6.2, 64 bit<br>CentOS 6.4 and 6.2, 64 bit | 2.1.1 | 1.1.4 |
| PXF | Extensibility layer to provide support for external data formats such as HBase and Hive. | Windows 2008 RedHat 6.4 and 6.2, 64 bit<br>CentOS 6.4 and 6.2, 64 bit | 2.1.1 | 1.1.4 |
| Pivotal HD | Pivotal Hadoop | Windows 2008 RedHat 6.4 and 6.2, 64 bit<br>CentOS 6.4 and 6.2, 64 bit | 1.1.1 | 1.1.4 |
| Pivotal Data Loader | Data Loader is a management tool that loads data to distributed data analytics platforms such as Hadoop, and HAWQ. | Windows 2008 RedHat 6.4 and 6.2, 64 bit<br>CentOS 6.4 and 6.2, 64 bit | 2.0.4 | 1.1.4 |

# 7 PADS 1.1.4 and Pivotal HD Documentation

The following PADS and related documentation is available in PDF format on our website at www.gopivotal.com.

HTML versions of our documentation is available here: docs.gopivotal.com/pivotalhd/

Additionally, you can still access product documentation from EMC's Support Zone.

**Table: HAWQ Documentation**

| Title | Revision |
|---|---|
| Pivotal ADS 1.1.4 Release Notes (This document) | A02 |
| Pivotal HAWQ 1.1 Installation Guide | A11 |
| Pivotal HAWQ 1.1 Administrator Guide | A08 |
| Pivotal HD Enterprise 1.1 Installation and Administrator Guide | A03 |
| Pivotal HD DataLoader 2.0 Installation and User Guide | A06 |
| Pivotal HD 1.1 Stack and Tool Reference Guide | A02 |
| Pivotal Command Center 2.1 User Guide | A02 |
| Pivotal Extension Framework Installation and User Guide | A02 |

# 8 Upgrading to HAWQ 1.1.x

The upgrade path supported for this release is HAWQ 1.0.x to HAWQ 1.1.x.

For detailed upgrade procedures and information, see the following sections:

- Upgrading from HAWQ 1.1.x to HAWQ 1.1.y
- Upgrading from 1.0.x to HAWQ 1.1.x

**Note:** Pivotal recommends that you back up any existing data before upgrading to HAWQ1.1.x.

**Note**:Follow these instructions if you installed HAWQ manually. To upgrade PHD Manager, see the Pivotal HD Enterprise 1.0 Installation and Administration Guide.

## 8.1 Upgrading from HAWQ 1.1.x to HAWQ 1.1.y

An upgrade from HAWQ 1.1.x to HAWQ 1.1.y involves stopping HAWQ, updating the HAWQ software binaries, and restarting HAWQ.

Log in to your HAWQ master host as the HAWQ administrative user:

```
$ su - gpadmin
```

Perform a smart shutdown of your current HAWQ 1.1.x system (shut down all active connections to the database):

```
$ gpstop
```

Run the installer for 1.1.y on the HAWQ master host using rpm. This installs HAWQ to /usr/local/hawq-1.1.y alongside any older versions, and it will point a soft link from /usr/local/hawq to /usr/local/hawq-1.1.y.

```
$ su - root # rpm -ivh hawq-1.1.y.x86_64.rpm --force
```

Run the following command to install the HAWQ 1.1.y binaries on all the hosts specified in the *hostfile* :

```
# gpssh -f hostfile -e "rpm -ivh hawq-1.1.y.x86_64.rpm --force"
```

After all segment hosts have been upgraded, you can log in as gpadmin user and restart your HAWQ system:

```
$ su - gpadmin
$ gpstart
```

## 8.2 Upgrading from HAWQ 1.0.x to HAWQ 1.1.x

This section describes how you can upgrade from HAWQ 1.0.x or later to HAWQ 1.1.x.

This section divides the upgrade into the following phases: pre-upgrade preparation, software installation, upgrade execution, and post-upgrade tasks.

**Important**: Carefully evaluate each section and perform all the required and conditional steps. Failing to perform any of these steps can result in a aborted upgrade, placing your system in an unusable or even unrecoverable state.

## 8.3 Preupgrade Preparation

Perform these steps on your current HAWQ system. This procedure is performed from your HAWQ master host and should be executed by the HAWQ superuser (gpadmin).

Log in to the HAWQ master as the gpadmin user

(*optional* ) Vacuum all databases prior to upgrade. For example:

(*optional* ) Clean out old server log files from your master and segment data directories. For example, to remove log files from 2011 from your segment hosts:

Note: Running Vacuum and cleaning out old logs files is not required, but it will reduce the size of HAWQ files to be backed up and migrated.

Run gpstate to check for failed segments.

If you have failed segments, you must recover them using gprecoverseg before you can upgrade.

Copy or preserve any additional folders or files (such as backup folders) that you have added in the HAWQ data directories or $GPHOME directory. Only files or folders strictly related to HAWQ operations are preserved by the migration utility.

## 8.4 Install the HAWQ Software Binaries

Run the installer for 1.1.x on the HAWQ master host using rpm. This installs HAWQ to /usr/local/hawq-1.1.x alongside any older versions, and it will point a soft link from /usr/local/hawq to /usr/local/hawq-1.1.x.

Run the following command to then install the HAWQ 1.1.x binaries on all the hosts specified in the *hostfile.*

# 9 Upgrade Execution

During upgrade, all client connections to the master are locked out. Before performing this procedure, inform all database users of the upgrade and lockout time frame. From this point onward, be sure that no one is on the system until the upgrade is complete.

1. Source the path file from your old 1.0.x installation. For example:

   ```
   $ source /usr/local/hawq-1.0.x/greenplum_path.sh
   ```

2. If your system has a standby master host configured, remove the standby master from your system configuration. For example:

   ```
   $ gpinitstandby -r
   ```

3. Perform a clean shutdown of your current ADS system. For example:

   ```
   $ gpstop
   ```

4. Source the path file from your new HAWQ 1.1.x installation. For example:

   ```
   $ source /usr/local/hawq-1.1.x/greenplum_path.sh
   ```

5. As gpadmin, run the 1.1.x version of the migration utility specifying your old and new GPHOME locations. If your system does not have mirrors, use gpmigrator. For example on a system with mirrors:

   ```
   $ su - gpadmn
   $ gpmigrator /usr/local/hawq-1.0.x /usr/local/hawq-1.1.x
   ```

   Note: If the migration does not complete successfully, contact Customer Support. See Troubleshooting a Failed Upgrade.
6. The migration can take a while to complete. After the migration utility has completed successfully, the HAWQ 1.1.x system will be running and accepting connections.

## 9.1 Post Upgrade on your HAWQ 1.1.x System

1. If your system had a standby master host configured, reinitialize your standby master using gpinitstandby:

   ```
   $ gpinitstandby -s standby_hostname
   ```

2. If your system uses external tables with gpfdist, stop all gpfdist processes on your ETL servers and reinstall gpfdist using the compatible HAWQ 1.1.x Load Tools package. Application Packages are available at the EMC Download Center.
3. If you want to use the Pivotal Command Center management tool, install the latest Command Center Console. To update your environment variable to point to the latest Command Center binaries, source the gpperfmon_path.sh file from your new installation.

Note: The Pivotal Command Center management tool replaces Greenplum command Center and Greenplum Performance Monitor. Command Center Console packages are available from the EMC Download Center.

4. If you had created tables using GPXF, you need to DROP these tables and CREATE them again using PXF 2.0.1.

See Appendix G, Pivotal Extension Framework, in the Pivotal ADS 1.1 Administrator Guide, A02, for instructions about how to install PXF 2.0.1.

Note: When you CREATE the tables for PXF2.0.1, remember to perform the following:

a. Change the protocol name in the LOCATION clause from gpxf to pxf.

b. Ensure that Fragmenter, Accessor, and Resolver are *always* specified for the table.

c. Manually recreate the two symbolic links that are removed during the update:

```
cd /usr/lib/gphd # base directory for PHD
mkdir publicstage
chown hdfs /usr/lib/gphd/publicstage
chmod 777 /usr/lib/gphd/publicstage
ln -s gpxf-x.x.x gpxf
```

d. Check that you have the new names for the Fragmenter, Accessor, and Resolver classes.

See Appendix G, Pivotal Extension Framework, in the *PADS 1.1 Administrator Guide* , A02, for information about the Java classes.

e. Check that you are using the correct gucs for PXF:

```
gpxf_enable_filter_pushdown -> pxf_enable_filter_pushdown
gpxf_enable_stat_collection -> pxf_enable_stat_collection
gpxf_enable_locality_optimizations -> pxf_enable_locality_optimizations
```

5. Use the pgcrypto package from 1.1.3 distribution and reinstall pgcrypto after the upgrade is done. Follow the instructions in the Pivotal HAWQ Installation Guide to install pgcrypto.

6. Install the PL/R language package according to instructions in the *Pivotal HAWQ Installation Guide*.

7. Inform all database users of the completed upgrade. Tell users to update their environment to source the HAWQ 1.1.x installation (if necessary).

# 10 Troubleshooting a Failed Upgrade

If you experience issues during the migration process, go to the Support page at Support Zone or contact Pivotal customer support at one of the following numbers:

- United States: 800-782-4362 (1-800-SVC-4EMC)
- Canada: 800-543-4782
- Worldwide: +1-508-497-7901

Be prepared to provide the following information:

- A detailed list of upgrade procedures completed.
- Log output from gpmigrator (located in ~/gpAdminLogs).

.