

Private Protocol is Dead! Long Live DRDA!

Hugh Smith (smithhj@us.ibm.com)
IBM

August 2, 2010
7983



SHARE in Boston

Agenda

- To Be or Not To Be Private Protocol (a historical perspective)
 - Private protocol deprecated in every version since V5
- What is alias resolution?
- Private Protocol Elimination First Steps (V9)
- V10 Private Protocol Elimination
- Private Protocol to DRDA Protocol Conversion Aids

To Be or Not To Be Private Protocol?

Timeline

Version	Event
2.2	Private protocol introduced; triggered by 3-part name reference
2.3	DRDA protocol and packages introduced; triggered by CONNECT TO statement
5	TCP/IP introduced and DRDA can only utilize
6	DRDA 3-part name support introduced with DBPROTOCOL bind option and all migrated plans and packages from V5 became DBPROTOCOL(PRIVATE)
9	DBPROTCL ZPARM removed; DRDA is default value for DBPROTOCOL; warning if PRIVATE used or implied
10	Private protocol eliminated; DRDA enhanced with alias resolution

To Be or Not To Be Private Protocol?

- Why should Private Protocol be eliminated?
 - Works Poorly
 - Error Prone
 - Not enhanced since DB2 V4
 - Can only use SNA communications
 - Can only access other DB2 for z/OS subsystems/groups
- Benefits of using DRDA protocol include
 - Specifically designed for heterogenous networks
 - Seamless Database Communications
 - Robust Performance
 - Max Block Extra and Multi-row Fetch and Insert support

To Be or Not To Be Private Protocol? ...

- DRDA protocol requires packages be bound at each site
 - Bind prior to execution results in more “predictable” and better performance (no incremental bind costs)
 - Original DB2 for z/OS requester PLAN requires remote packages be specified in PKLIST
 - ROT: Always define PKLISTs using an ‘*’ for location names in the entries, such that every time a new location could be accessed by an application using the PLAN, all that is required is a BIND PACKAGE COPY of the local package at the new remote location.
- DRDA 3-part name alias support at a DDF requester did not resolve alias references prior to sending request to server
 - Private protocol did perform alias resolution
 - Some private protocol scenarios could not be implemented with DRDA

What is Alias Resolution?

- Example of 3-part name alias
 - CREATE ALIAS TEST1 FOR LOC1.AUTHID2.TABLE3
 - Important to note that alias created under AUTHID1
- Example statement requested under AUTHID1:
 - SELECT * FROM TEST1
- Private protocol processing of statement:
 - Requester catalog is used during optimization
 - ALIAS AUTHID1.TEST1 found defining AUTHID2.TABLE3 at location LOC1
 - Substitute (resolve) alias reference in statement
 - SELECT * FROM "AUTHID2"."TABLE3" is sent to LOC1

What is Alias Resolution? ...

- DRDA protocol processing of statement:
 - Requester catalog is used during optimization
 - ALIAS AUTHID1.TEST1 found defining AUTHID2.TABLE3 at location LOC1
 - SELECT * FROM TEST1 sent to LOC1 during PREPARE/EXECUTE IMMEDIATE or during remote package bind of statement
 - 2-part name alias at LOC1 may be required, i.e. CREATE ALIAS AUTHID1.TEST1 FOR AUTHID2.TABLE3

Private Protocol Elimination First Steps

- DB2 9 for z/OS initially delivered:
 - Warning issued when DBPROTOCOL(PRIVATE) specified or implied at BIND time
 - Message DSNT226I
 - Prevent PRIVATE being the default value for the DBPROTOCOL bind option (DBPROTCL ZPARM)
 - IFCID157 trace can be used to determine applications using private protocol
 - Analysis Tool to identify and convert plans and packages
 - DSNTDP2DP with install job DSNTIJPB
 - Additionally identify aliases required at remote site

Private Protocol Elimination Next Steps

- V8|V9 enhancements to aid conversion to DRDA protocol
 - Enable or disable private protocol (PK92339)
 - Via ZPARM PRIVATE_PROTOCOL (default YES)
 - Enables testing to ensure private is eliminated
 - Enable or disable DRDA alias resolution processing (PK64045 + PM14816(V9))
 - Via ZPARM DRDA_RESOLVE_ALIAS
 - Provide Enhanced Tools to identify plans and packages requiring conversion (PK64045)
 - DSN1PPTP analyses IFCID157 trace records for input into DSNTTP2DP
 - DSNTTP2DP generates commands to convert packages or plans to DRDA
 - DSNTPPCK and install job DSNTIJPCK to check for future invalid statements

V10 Private Protocol Elimination Overview

- No inbound or outbound of private protocol supported
- Removal of capabilities that applied to private protocol
 - Prevent DBPROTOCOL(PRIVATE)
 - Message DSNT225I or SQLCODE -30104 issued
 - Certain SQL syntax not supported
- DBPROTOCOL(DRDA) will be assumed for any BIND, REBIND, or BIND PACKAGE COPY operation regardless of existing package's DBPROTOCOL value
- Support co-existence with V8|V9 members in data sharing group while DB2 10 in CM8 or CM9
 - Allows pre-V10 private protocol requesters to continue to access data in group while V8|V9 members still exist
 - LULIST must be used to access group

V10 Private Protocol Elimination Overview

- Any package or plan last bound with DBPROTOCOL(PRIVATE) will be run under DBPROTOCOL(DRDA) processing rules
 - Packages and plans will be allocated successfully if only local objects accessed
 - Any package which attempts remote access will receive resource unavailable due to reason 00E3001E
 - Package must be converted to use DRDA protocol
- DRDA protocol will always perform alias resolution in V10
 - Dynamic SQL PREPARE or EXECUTE IMMEDIATE request
 - Remote BIND PACKAGE

Invalid Private Protocol Request to V10

- DDF response from V10 subsystem or member:
 - Reject request
 - VTAM sense code '10086021'
 - Requestor (pre-V10 system) will receive
 - -904 Reason Code '00D31026' Type '1001'

Invalid SQL Statements

- The following SQL syntax will no longer be supported:
 - “AT ALL LOCATIONS” on GRANT/REVOKE of table permissions
 - “ALL PRIVATE” on RELEASE statement
 - Binds will fail for any packages containing statements with the above syntax
 - Dynamic SQL will receive -084
- V8|V9 REXX Tool DSNTPPCK
 - Syntax:
`DSNTPPCK SSID=ssnm [SYSPACKSTMT=N|Y SYSSTMT=N|Y]`
 - Scans SYSIBM.SYSPACKSTMT and/or SYSIBM.SYSSTMT looking for above SQL syntax in statically bound SQL statements
 - Run before migrating to V10
- SYSTABAUTH and SYSCOLAUTH catalog tables
 - PUBLIC* treated as PUBLIC in CM8|CM9
 - PUBLIC* changed to PUBLIC during ENFM

V10 DRDA Alias Resolution

- Dynamic SQL Example:
 - Under AUTHID of SMITH, issue:
`CREATE ALIAS MYALIAS1 FOR STLEC2.TEST.DEPT`
 - Via DSNTEP2|4 or SPUFI and CURRENT SQLID of SMITH or primary authid of SMITH, issue:
`SELECT COUNT(*) FROM MYALIAS1`
 - DB2 will send to STLEC2:
`SELECT COUNT(*) FROM "TEST"."DEPT"`
 - Alias at remote location is not necessary unless that location wants to make the object remote from it
 - SQL statement will still be checked at requester for validity, i.e. all table/alias objects must reference a single location (SQLCODE -512)

V10 DRDA Alias Resolution ...

- Remote package bind example:
 - Using same alias as dynamic example, program XYZ has an embedded static SQL statement:
`EXEC SQL SELECT COUNT(*) INTO ... FROM MYALIAS1`
 - Program XYZ is precompiled and bound locally via:
`BIND PACKAGE(COLLID1) MEMBER(XYZ) QUALIFIER(SMITH) ...`
 - Package XYZ is bound at STLEC2 via:
`BIND PACKAGE(STLEC2.COLLID1) MEMBER(XYZ) QUALIFIER(SMITH) ...`
or
`BIND PACKAGE(STLEC2.COLLID1) COPY(COLLID1.XYZ) QUALIFIER(SMITH) ...`
 - Requester's catalog accessed to resolve any alias in statement but will not check for errors
 - Modified SQL statement sent to STLEC2 to be bound in target package:
`SELECT COUNT(*) FROM "TEST"."DEPT"`
 - Package at remote system has the modified SQL statement while requester system package has application's original SQL statement

V10 DRDA Alias Resolution ...

- Remote BIND PACKAGE rules:
 - Specification of QUALIFIER/OWNER governs how requester's catalog searched to resolve aliases of unqualified object references
 - QUALIFIER value used if both OWNER/QUALIFIER specified
 - OWNER value used if QUALIFIER not specified but will be modified by any outbound userid translation specified in CDB
 - Bind job userid will be used if QUALIFIER and OWNER not specified but will be modified by any outbound userid translation specified in CDB

V10 DRDA Alias Resolution Observations

- Initial application package should be bound locally via member DBRM at any requester location
 - Package will contain original text of application statements
 - Distributed clients should bind their package at any location they directly access
- Requester package should then be copied to any location immediately accessed by requester application
 - Use remote BIND PACKAGE COPY
 - Package copies at remote locations contain modified SQL statements
 - If SQL statements in remote location package copies contain 3-part name alias references, package at remote location should be copied from remote location to locations directly accessed via remote site
- Mixing of CONNECT TO statements and 3-part name alias references in single package should be avoided
 - Local BIND of member DBRM at remote location accessed by CONNECT TO will have to be utilized since alias resolution will always be performed during remote BIND PACKAGE from requester location
 - No determination at BIND time can be made as to when a statement in a package is going to be used to access local location, CONNECT TO location, or location accessed from local location via 3-part name alias reference

Private Protocol to DRDA Conversion Aids

- DSNTDP2DP – Private to DRDA protocol catalog analysis program:
 - Syntax: `DSNTDP2DP` `SSID=ssnm`
`[PACKAGES=Y|N]`
`[PLANS=Y|N]`
`[PPINTOTABLE=table_name]`
 - Generate commands for all private protocol objects even those which do not have a remote location dependency:
 - When locations cannot be detected, dummy commands will be generated with comments on how user should proceed
 - PLANS option will generate REBIND PLAN() COLLID(*) commands
 - PLANS option will generate REBIND PLAN commands for any PLAN defined with blank location-name values in any entry of its PKLIST (an asterisk (“*”) will be substituted)
 - PPINFOTABLE option table, if specified, will be searched for any serving location information for packages and plans
 - Install job, DSNTIJPDP, can be tailored to run program

Sample OUTPKGS Generated Output

```
REBIND PACKAGE(DSNESPCS.DSNESM68) -
  DBPROTOCOL(DRDA) PLANMGMT(OFF)
* DSNTDP2DP could only determine locations for this package
* from the private protocol information table.
BIND PACKAGE(STLEC4B.DSNESPCS) COPY(DSNESPCS.DSNESM68) -
  OPTIONS(COMPOSITE) -
  OWNER(SYSADM) QUALIFIER(SYSADM) -
  DBPROTOCOL(DRDA) SQLERROR(CONTINUE)
REBIND PACKAGE(DSNESPRR.DSNESM68) -
  DBPROTOCOL(DRDA) PLANMGMT(OFF)
* DSNTDP2DP could only determine locations for this package
* from the private protocol information table.
BIND PACKAGE(STLEC4B.DSNESPRR) COPY(DSNESPRR.DSNESM68) -
  OPTIONS(COMPOSITE) -
  OWNER(SYSADM) QUALIFIER(SYSADM) -
  DBPROTOCOL(DRDA) SQLERROR(CONTINUE)
REBIND PACKAGE(DSNESPUR.DSNESM68) -
  DBPROTOCOL(DRDA) PLANMGMT(OFF)
* DSNTDP2DP could only determine locations for this package
* from the private protocol information table.
BIND PACKAGE(STLEC4B.DSNESPUR) COPY(DSNESPUR.DSNESM68) -
  OPTIONS(COMPOSITE) -
  OWNER(SYSADM) QUALIFIER(SYSADM) -
  DBPROTOCOL(DRDA) SQLERROR(CONTINUE)
```

```
REBIND PACKAGE(DSNTEP2.DSNTEP2) -
  DBPROTOCOL(DRDA) PLANMGMT(OFF)
* DSNTDP2DP could only determine locations for this package
* from the private protocol information table.
BIND PACKAGE(STLEC4B.DSNTEP2) COPY(DSNTEP2.DSNTEP2) -
  OPTIONS(COMPOSITE) -
  OWNER(SYSADM) QUALIFIER(SYSADM) -
  DBPROTOCOL(DRDA) SQLERROR(CONTINUE)
REBIND PACKAGE(TESTCOLLID.LI682C) -
  DBPROTOCOL(DRDA) PLANMGMT(OFF)
BIND PACKAGE(STLEC4B.TESTCOLLID) COPY(TESTCOLLID.LI682C) -
  OPTIONS(COMPOSITE) -
  OWNER(SYSADM) QUALIFIER(ADMFO01) -
  DBPROTOCOL(DRDA) SQLERROR(CONTINUE)
REBIND PACKAGE(TESTCOLLID.LI682D.(V1R1M0)) -
  DBPROTOCOL(DRDA) PLANMGMT(OFF)
BIND PACKAGE(STLEC4B.TESTCOLLID) COPY(TESTCOLLID.LI682D) -
  COPYVER(V1R1M0) OPTIONS(COMPOSITE) -
  OWNER(SYSADM) QUALIFIER(ADMFO01) -
  DBPROTOCOL(DRDA) SQLERROR(CONTINUE)
```

Sample OUTPLANS Generated Output

```
REBIND PLAN(LI682A) COLLID(*) DBPROTOCOL(DRDA)
REBIND PLAN(LI682A) -
  PKLIST(*.DSN_DEFAULT_COLLID_LI682A.* -
    )
BIND PACKAGE(STLEC4B.DSN_DEFAULT_COLLID_LI682A) -
  COPY(DSN_DEFAULT_COLLID_LI682A.LI682A) -
  COPYVER(V1R1M1) OPTIONS(COMPOSITE) -
  OWNER(SYSADM) QUALIFIER(ADMF001) -
  DBPROTOCOL(DRDA) SQLERROR(CONTINUE)
BIND PACKAGE(STLEC4B.DSN_DEFAULT_COLLID_LI682A) -
  COPY(DSN_DEFAULT_COLLID_LI682A.LI682D) -
  COPYVER(V1R1M0) OPTIONS(COMPOSITE) -
  OWNER(SYSADM) QUALIFIER(ADMF001) -
  DBPROTOCOL(DRDA) SQLERROR(CONTINUE)
REBIND PLAN(LI682B) COLLID(*) DBPROTOCOL(DRDA)
REBIND PLAN(LI682B) -
  PKLIST(*.DSN_DEFAULT_COLLID_LI682B.* -
    , *.TESTCOLLID.LI682D -
    )
BIND PACKAGE(STLEC4B.DSN_DEFAULT_COLLID_LI682B) -
  COPY(DSN_DEFAULT_COLLID_LI682B.LI682B) -
  COPYVER(V1R1M2) OPTIONS(COMPOSITE) -
  OWNER(SYSADM) QUALIFIER(ADMF001) -
  DBPROTOCOL(DRDA) SQLERROR(CONTINUE)
* The following REBIND PLAN commands are needed to
* convert DBPROTOCOL(PRIVATE), pklist-only PLANS (where
* all PKLIST entries have blank or '*' locations) to
* DBPROTOCOL(DRDA) as well as adjusting their pklist
* entries to support DBPROTOCOL(DRDA) (location='*').
REBIND PLAN(LI682C) DBPROTOCOL(DRDA) -
  PKLIST(*.TESTCOLLID.LI682C -
    , *.TESTCOLLID.LI682D -
    )
```

* The following REBIND PLAN commands are needed to
 * update the PKLISTS of DBPROTOCOL(DRDA), PKLIST-only
 * plans by substituting an asterisk ('*') for a blank
 * location name. The rebind may be necessary if any
 * package that could be accessed by the plan was
 * previously private protocol and this tool generated
 * the commands to convert it to DRDA.

```
REBIND PLAN(DSNEDCL) -
  PKLIST(*.DSNEDCL.DSNECP68 -
    , *.DSNTIAP.DSNTIAP -
    )
REBIND PLAN(DSNESPCS) -
  PKLIST(*.DSNESPCS.DSNESM68 -
    , *.DSNTIAP.DSNTIAP -
    )
REBIND PLAN(DSNESPRR) -
  PKLIST(*.DSNESPRR.DSNESM68 -
    , *.DSNTIAP.DSNTIAP -
    )
REBIND PLAN(DSNESPUR) -
  PKLIST(*.DSNESPUR.DSNESM68 -
    , *.DSNTIAP.DSNTIAP -
    )
REBIND PLAN(DSNHYCRD) -
  PKLIST(*.DSNHYCRD.DSNHYCRD -
    )
REBIND PLAN(DSNWZP) -
  PKLIST(*.DSNWZP.DSNWZP -
    )
```

Private to DRDA Protocol Conversion Aids

- DSN1PPTP – private protocol trace formatting program
 - Program will take input dataset which is a copy of all SMF 102 records generated due to started IFCID157 trace:
`-START TRACE(PERFM) CLASS(30) IFCID(157) DEST(SMF)`
 - Program will format all IFCID157 outbound private protocol trace records into a file which can be used by the load utility
 - Documentation will describe how to create a private protocol information table and load it with data from DSN1PPTP
 - Install job, DSNTIJPT, provided
- IFCID157 trace, DSN1PPTP, and creation of private protocol information table need only be performed if packages/plans have dynamic SQL
 - ROT: BIND packages with dynamic SQL at all remote locations accessible from this requesting location

Migrating to V10

- Which plans and/or packages must be converted to DRDA?
 - Run V10 pre-migration job DSNTIJPM (V8/V9 – DSNTIJPA)
 - Queries catalog to identify any objects requiring conversion
 - PLANs identified in REPORT19 output may experience resource unavailable due to reason 00E3001E and may required conversion to DRDA
 - PACKAGEs identified in REPORT21 output may experience resource unavailable due to reason 00E3001E and may required conversion to DRDA
 - Objects identified in REPORT18 or REPORT21 need not be rebound

Steps for Converting to DRDA

1. Activate IFCID157 trace to capture locations accessed
2. Run tailored job, DSNTIJPT (DSN1PPTP), to create private protocol information table from IFCID157 trace data (optional if step 1 not performed)
3. Run tailored job, DSNTIJPD (DSNTP2DP), to generate PLAN/PACKAGE management commands
4. Enable DRDA Alias Resolution if V8 or V9 (DRDA_RESOLVE_ALIAS ZPARM set to YES)
5. Run generated commands tailored to your installation's requirements
6. Test applications which use converted PLANs/PACKAGEs
 - Check for errors from DRDA conversion
 - SQLCODE -805 suggests embedded dynamic SQL
 - Fix using BIND PACKAGE COPY of indicated package

Summary

- Private protocol no longer supported in V10
 - Not all packages have to be rebound with DBPROTOCOL(DRDA)
 - (see output from DSNTIJPM (DSNTIJPA))
- DRDA protocol has been enhanced to perform alias resolution in V10
 - Optional in V8/V9 but recommend you must eventually enable prior to migration to V10
- Tools/aids provided to assist in converting to DRDA