Data Base Management

DB2 Application Development Reference Guide

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Reference Guide

<u>FOREWORD</u>

In the traditional development environment, projects are often started at the analysis or design levels. All too many times, these projects are begun to satisfy the perceived need to have a file, which contains certain information, or to have a program, which can accomplish certain tasks.

The problem with projects, which start with these Objectives, has been the failure to fully address or to fully communicate the business requirements or needs which are behind the development requests. Assumptions about the client's business needs, or assumptions about the final capabilities of a software program, often lead to the development of systems which take longer to develop than is initially estimated, which costs more than is originally budgeted, and which do not provide the full information or functionality that is originally envisioned by both the client and the developer.

To address these shortcomings with traditional development, methodologies have been defined which provide a way of defining, scoping, and accomplishing projects to best meet the business requirements of the client areas. While the methodology provides a guide to the development process, and while DBMS tools are available to help developers progress through the methodology processes, there is still a need for standards and procedures to define the specific development requirements of the individual data processing environment.

Standards and procedures provide a framework of actions and methods by which both project development and communication are universally accomplished. This framework provides both a list of deliverables for the development staff to accomplish, as well as, a list of expectations upon which all interested parties may act. While many of these tasks may seem trivial, and others may seem cumbersome or involved, each contributes to a more complete project plan, better communication and planning, and, therefore, to the more likely chance of project success.

Office of the CIO (OCIO) and the many agencies that OCIO serves do not have universal standards and procedures. Rather, development standards and procedures are generally maintained at the agency level, while performance standards and operational procedures are maintained at the OCIO level.

Reference Guide

OCIO defines standards and procedures in the following way:

STANDARD

"A standard is a yard stick against which any task can be measured to assure it has been completed using the correct steps and procedures."

PROCEDURE

"A description of the steps to be followed, in a regular and specified order, to carry out the tasks required by a standard or those recommended in a guideline."

The following DB2 Application Development reference information is not meant to be all-inclusive. It is intended to be used as a framework for DB2 applications development as that development relates to an operation within 'OCIO'' central data base management systems. The guidelines and information are provided in anticipation of an expected set of events, and an expected set of behaviors, which help all persons involved in a development effort to know what is required at all phases of the development cycle.

The following goals and operation actions are presented as the minimum framework to be used for both traditional and IE project development efforts. No effort has been made, however, to declare this information as fully comprehensive or complete. Where additional goals or operation actions are desired or required, they will be added to the document. Where they need to be changed, changes will be considered.

Reference Guide

<u>*Definitions Page*</u>

When describing various terms the following definitions will be used:

Agency Adm – Considered the owner of an object (e.g. table, bind qualifier).

Application – The programs that are used to accomplish a specific task.

Application Group – Responsible for the building and setup and maintenance of applications to execute a series of programs that are used to meet a Project's goals.

Project Manager – For Project being developed by/for a State Agency. They oversee and supervise the execution of the project to be sure it is done correctly and meets the needs of the Agency Administrator/Agency.

Project DBA – Includes Application DBA - they work with State Agencies for their Application Development needs and daily maintenance. They design and work with databases to meet an end user's need.

System DBA – Technology Support Services Group that is responsible for installation and maintenance of mainframe DB2 and associated products.

User - One for whom the project is built

PARTICIPATION IN PHASES OF DEVELOPMENT

Objective: System DBA/Project DBA will participate in all development phases of a project.

1. OCIO Management will assign an appropriate number of trained and experienced staff to a project to provide data base knowledge and support to that project.

2. Project Management will schedule projects, which involve System DBA/Project DBA staff, with sufficient lead-time to arrange staff schedules as required.

3. Senior OCIO Management will be the final authority regarding project priorities and/or scheduling conflicts.

4. Conflicts in priorities or scheduling will not be reason to circumvent or ignore any of the standards or procedures presented in this document.

FEASIBILITY (also planning)

Objective: System DBA/Project DBA will provide consultation services in the feasibility or planning phase of a project.

1. All projects must be evaluated based upon their importance or their payback in the framework of the Enterprise model. The feasibility of a project is normally determined within the dictates of the Integrated Strategic Plan, and should include the reason or the payback, which gives the project its importance.

2. Traditional projects typically are based upon a more direct and immediate payback from the project itself, rather than on an integrated benefit derived from its importance to the Enterprise. The System DBA/Project DBA group will participate in this phase and will provide good faith estimates for development time and effort based upon available information. These estimates can and should be used in preparing project justification.

3. In the event that business or regulatory requirements dictate the initiation of a project, without payback or cost justification, the Project DBA group should be consulted to accomplish the following two Objectives:

   a. The Project DBA group will have the information necessary to arrange schedule-time and manpower availability to support the project.

   b. The Project DBA group can provide a gross estimate of time and manpower needed to accomplish data base related tasks.

Reference Guide

Business Area Analysis

Objective: At least one Project DBA staff person will be attached to a project during the Business Area Analysis phase and will review all deliverables from this phase.

1. In the Information Engineering Methodology (IEM), the Business Area Analysis part of a project is completed in one phase (small and traditional projects), or in two phases in larger projects (Outline and Detailed BAA's). This phase defines the business requirements and rules, which will dictate the necessity for the project. It is this area analysis around which the system is designed.

2. The Project DBA will attach at least one person to the project during this phase. The Project DBA will be included in all 'reviews' conducted in this phase, which will ensure a complete understanding of the business requirements being addressed. This phase participation will further serve the effort to understand the requirements of the project, as well as to more fully appreciate the volumes and/or criticality of processes and data, which will ultimately be included in the construction of the project.

3. The Analysis project team will produce the following deliverables for Project DBA review:

   a. A complete Entity Relationship Diagram (ERD). This diagram will normally show all entity types and relationships in third normal form. Denormalization for performance will be addressed during the design phase. All 'Many to Many' relationships should be resolved to facilitate the naming of resultant associative entity types.

   b. Include with the ERD descriptions of entities and attributes. An example of VENDOR entity would be 1 and only 1 VENDOR can have 1 to M vehicles.

   c. When possible, Process Logic Diagrams (PLD's) should be created. These diagrams help depict the Process and Data interactions of the business. While the formal methodology suggests that individual Process Logic Diagrams be prepared for each elementary process, grouped notations on a copy of the ERD will, in many cases, satisfy the intent of this documentation.

4. In all cases, physical DB2 objects will be created only after the Project DBA had reviewed the objects and has had sufficient opportunity to understand the data and relationships in the system or subject area. The review will normally be conducted by the Project DBA staff member attached to the project. This presentation will help insure that the staff member has sufficient understanding of the system or subject area and that the documentation supports the facts as they are presented. Under normal circumstances, no more than twenty (20) tables and relationships will be presented at one time, and no more than 2 reviews will be conducted during a one week time period.

Reference Guide

<u>*BUSINESS SYSTEM DESIGN*</u>

Objective: At least one Project DBA staff person will be assigned to a project during the Business System Design phase. The Project DBA member will review all models and diagrams and gain a thorough understanding of the proposed system processing and the evolving data base objects.

1.   Project DBA will provide the project with the technical design of its data base system.

   - A staff member will be assigned to the project.

   - The entities will be translated to DB2 DDL. This phase will mark the first intensive participation by the DBA.

   - The Analysis project team will use the design tools to remove errors and will generate the DDL to be ported to the mainframe.

   - Errors found during this phase will be corrected by the design team. If errors are substantial, the translation process will be repeated after the error correction process.

2.   The physical objects defined in the DBMS environment will be a test project database, tablespaces, tables, and all indexes. The design team will provide the DDL along with the following physical design estimates:

   Initial volume and growth of each entity:
   a.   Estimate of number of rows for each table
   b.   Growth statistics (number of rows that will be added in the future).
   c.   Growth horizon (how often the amount of "Growth Statistics" will be added (i.e. 2,000 rows added every year).

Reference Guide

CONSTRUCTION

Objective: At least one Project DBA staff person will be assigned to a project during the Construction phase. The Project DBA staff member will provide technical assistance to the development staff in both their programming efforts and in their testing environments.

1. In general, all program SQL and DML will be designed and coded in the most resource efficient manner possible.

2. No application program will be designed nor coded using dynamic SQL, unless reviewed and approved by System DBA.

3. Individual program SQL code may be subject to review and approval by Project DBA.

4. At the discretion of Project DBA, the program may be subject to monitor and/or review in a CAT/Volume environment to ensure program execution efficiency.

5. All batch programs will use checkpoints or commit points in their processing with the following exceptions:

    ▪ Read only programs, which are estimated to run less than 15 minutes.
    ▪ Update programs, which are estimated to run less than 30 seconds.
    ▪ One-time, request, or conversion programs which are estimated to run less than 30 minutes, and for which an image copy is taken prior to the run.
    ▪ Any program, which for business or operational considerations, is approved by Data Base Management to run without issuing checkpoints and/or commit points.

6. All batch programs will contain appropriate code to ensure that they can be restarted from the last checkpoint or commit point.

7. Project DBA will promote all DB2 objects to the appropriate test levels.

8. Project Analysis team will be responsible for arranging the loading, restoring, or copying of data in all test environments.

9. All data used in a test environment must be either:

    ▪ Created in the test environment based upon fictitious persons, conditions, or combinations of data which do not resemble nor can be used to identify real persons or conditions in a department's production data base.
    ▪ Production data which has been sufficiently sanitized or altered to prevent the ready identification of persons or conditions by others not associated with the department nor its business.

Reference Guide

DB2 OBJECT NAMING CONVENTIONS

Objective: All DB2 database objects and all DB2 object qualifiers or creators will use a standard naming convention.

DB2 FILE STRUCTURE

DB2 tablespaces and index spaces use VSAM for the access method and are created as data sets with the following name format:

NAMING GUIDELINES: **sss.DSNDBD.ssstDBnn.sssTSnnn.I0001.A001**

sss	= VCAT Alias name All Test use DBT Production use System Acronym + '2' (e.g. RFS2, NFO2)
DSNDBD/C	= DB2 Data/Cluster designation
ssstDBnn	= Data Base name (see description below)
sssTSnnn	= Table space (see description below)
or	
iiiiiiii	= Index space (generated by DB2)
I0001	= literal (either I or J)
Annn	= Partition (A + partition number)

ACCESS to DB2 Linear dataset.

Access to the DB2 linear datasets is restricted through RACF. All Data access to DB2 will be routed through DB2 controls. Any exceptions will need to be justified and authorized by senior management of the relevant agencies and also by the Office of the CIO.
In production, RACF ALTER access must be granted to the following system and support groups based on the two high nodes of the DB2 data sets (i.e.   sss2.DSNDBC.*.**):

CDPPROD (only if you are running scheduled db2 utility jobs under this id)

xxxMSTR (and second data sharing member xxxxMSTR)

ex. DB2MSTR and DB2PMSTR.

xxxDBM1 (and second data sharing member xxxxDBM1)

ex. DB2DBM1 and DB2PDBM1

DBM0000 – DBM Group Superior RACF Group.

STORAGE GROUPS

Note: Starting in 2003 there has been effort to convert all db2 linear datasets from DB2-Managed to SMS/ACS Managed. All Project DBA's will work with System DBA and MVS Groups to have these routines set up.

To take advantage of SMS Managed, any specific hard-coded volume reference must be removed from your stogroup and you would add (ALTER or CREATE) '*' to the volume. (ie. Instead of **VOLUMES("339B53")** you would use **VOLUMES("*")**.

Then the MVS group manages which volumes get used by respective stogroups.

1. The default DB2 Storage Group SYSDEFLT will normally be used in the test system.

   a. In Test use of SYSDEFLT will be granted to PUBLIC.

   b. In Production (due to TEMP Declared Global Temporary Tables objects being moved to Workfile db (i.e. DSNDB07)) the use of SYSDEFLT will be granted to Agency Adm. and Project DBA group ids.

2. Special DB2 Storage Group(s) may be set up in test when the volume of test objects or charge-back considerations warrants their use.

3. DB2 Storage Groups will be used in the production system and will be named using the following naming convention:

   NAMING GUIDELINES**: xxxSTOGn**

Where:

   - xxx   = Department (System) Acronym
   - STOG   = Literal
   - n   = Number

Reference Guide

DB2 DATA BASE

1. The default DB2 database DSNDB04 <u>will not</u> be used by any project.

2. Project data bases will be named using the following naming convention:

   NAMING GUIDELINES**: xxxtDBnn**

   Where:

   xxx   = Department (System) Acronym

   t   = Tier
'U'	= Unit
'S'	= String
'C'	= CAT
'T'	= Training
'P'	= Production
'E'	= Exception (data base to hold all exception tables per system)
'F'	= Federal
'Q'	= QMF

   DB   = Literal,   D2 = Subsequent Test DB,   D3 = Subsequent Test DB

   nn   = data base version number or designation

'II'	= Original Production
'In'	= Subsequent Production data base*
nn	= test version** (or Path or User's initials)

*NOTE: The use of 'I1' should be avoided to help differentiate the database from the 'II' version.

**NOTE: Test version numbers 01 through 19 are normally reserved for system versions, with numbers 20 through 99 and letters AA through ZZ assigned to individuals.

    Only databases (and their associated objects) which have been assigned system test versions numbers will be migrated out of the unit test tier. Individuals may use their individual databases (and associated objects) as they determine necessary for their development and testing requirements. However, neither objects nor data, which exists in individual database(s) will be promoted to a higher tier of development.

    Test versions, which are intended to be used by batch processing or distributed applications only, will normally be assigned a version letter, instead of a number. This letter should correspond with the equivalent Tier and Chain of development.

DATABASE EXAMPLE

Development in the batch equivalent of the Chain A (Path 1), String Tier would have a database name 'xxxSDB0A'.

Development in the batch equivalent of the Chain B (Path 3), String Tier would have a database name 'xxxSDB0C'.

Test versions are intended to be used by both batch and online testing and will use the numeric version. Utility databases, developed and maintained in DB2 for use by Applications development (rather than the use by business applications) may use a less segmented naming convention. These databases will be identified and documented by the Project DBA.

TABLESPACES

Tablespaces will normally be defined as segmented and will normally contain one table. The tablespace name will be 8 bytes long and will use the following naming convention:

NAMING GUIDELINES**: xxxTSnnn** -- or -- **xxxTCnnn**

Where:	**xxx**	= Department (System) Acronym
	TS	= Literal for Source tablespace
	TC	= Literal for CD tablespace (SQL Replicator)
	nnn	= Unique combination of letters and/or numbers. This combination will normally be a number and will be the next available number.
xxxEEXPT		= Exception tablespace

TABLES

A table name may be any acceptable combination of letters and symbols for a DB2 table (see catalog table 'SYSIBM.SYSTABLES' name column for max length). It may not, however, end with the characters '_S', '_V', or '_E'. Those trailing characters are reserved for synonyms, views and exception tables respectively (see the following items). The same name will be used in all test and production environments.

*NOTE: Large agencies or applications should consider the use of a standard system prefix for each table. This prefix can help insure unique table names where the tables are not developed within a coordinated or central enterprise scheme.

INDEXES

An Index name may be any acceptable combination of letters and symbols for a DB2 index (see catalog table 'SYSIBM.SYSINDEXES' name column for max length). In general, a naming convention should be used which identifies the index as a primary and clustering index, a clustering and non-primary index, or a general support index.

NAMING GUIDELINES**: xxx?tttn**

Where:	**xxx**	= Department (System) Acronym
	?	= **I** for General support index,
		= **P** for primary and clustering index or primary index
		= **C** for clustering and non-primary index
	ttt	= Unique combination of letters and/or numbers.
	n	= a number or letter

xxx**I**0051 or xxx**I**005A   = the first <u>general index</u> for a table in tablespace '005'

xxx**P**0052 or xxx**P**005B   = the <u>primary</u> clustering index for a table in tablespace '005'

xxx**C**0061 or xxx**C**006A   = the <u>clustering</u> index for a table in tablespace '006' which is
                                             not a primary index.

SYNONYMS

Synonyms are unqualified pointers to physical tables. A synonym uses the CURRENT SQLID of the initiator to provide qualification. An SQL statement, which refers to a synonym, uses the unqualified single node name of the synonym. DB2 then, uses the current SQLID and the synonym name as its entry into the catalog table SYSIBM.SYSSYNONYMS, and uses the resultant TBCREATOR and TBNAME columns for the actual physical table name.

Synonyms were once used exclusively in application programs because table names were actually changed between the development and production environments. Where table names are not changed, there is no longer a need for this pointer capability to be used.

Synonyms should only be used when a table(s) from a different application system or with a different qualifier must be used. This situation typically occurs at the Unit or development level where the developer wants to mix tables and where he/she uses his/her own SQLID as a qualifier.

The use of synonyms in Legacy systems will continue to be supported. The synonyms are normally limited to 8 bytes, with the fourth position of the name using the letter 'S' to represent the letters 'T' or 'P' in the actual table name.

Unless the synonym is used in this manner, and in a Legacy system, the use of synonyms is discouraged. Rather, the developer is encouraged to use actual table names, or in some cases, ALIASES or VIEWS in his/her SQL statements.

VIEWS

Views are generally used when a subset of table columns, or a group of columns from two or more tables (JOIN), are to be referenced in a single object. In normal practice, Views are necessary only when data security concerns must be considered and enforced without restriction, or when certain tools require them to be able to construct code for JOIN operations. Views are limited to defining sets or subsets of tables, which exist in the same subsystem in which the view is defined.

Due to the specific nature of views, requests to create drop and access need to be submitted to the DBM Group for execution. For justification and further explanation please refer to the sections in the DB2 manuals that discuss Views.

If used, a view will be any allowable name followed by the characters '_V' or the same name as the table but a V in the fourth position.

COMMENT ON

You can add a comment to a <u>view</u> with the "COMMENT ON" statement, but you must have SYSADM or SYSCTRL authority. In SYSIBM.SYSTABLES, you'll see the result of this statement under the "<u>Remarks</u>" column.

To add a remark, you can create the view and send a request to the DBM Group to have the COMMENT ON statement run to add the remark. DBM Group can then run the statement against the view.

ALIASES

ALIASES are pointers to physical tables or VIEWS, which can have a different qualifier and/or LOCATION than the table or VIEW to which it points. They are generally used in Distributed Relational Database Architecture (DRDA) applications where access authority must be specifically defined for individual groups of users for a limited group of tables. They are generally preferred in applications where dynamic SQL must be used in a developed application and where DRDA protocols are utilized.

Unlike VIEWS, which require access GRANTS to be issued on the tables prior to the creation of the VIEW, access GRANTS on ALIASES may be made independently from the actual existence

of the table to which it points. Further, an ALIAS can be created, and will exist, whether or not the table to which it points actually exists at the time of creation.

The authority to access data is GRANTED on the underlying table rather than on the ALIAS itself. This security consideration differs markedly from the VIEW, which requires the creator of the VIEW to hold the authority on the underlying table, but who can pass on authority to access data from a table by GRANTING access to the VIEW, rather than granting access to the table itself.

If used, the Alias will use the same name as the table to which it points. No other modifier or designation is required.

Project DBA's can create an alias for applications they support if they have the DBADM authority. They cannot create an alias to other database tables, if they do not have DBADM authority.

EXCEPTION TABLES

A Check Data Utility exception table will be the same name as the table for which it provides exception processing, followed by the characters '_E'.

DCLGEN

1.   All DCLGEN's will be done on unqualified tables. Eight byte names will be assigned to the resulting Partitioned Data Set member for each of the tables. They will normally be the eight byte name used for the table's tablespace name, with the 'TS' replaced by the letters 'DG'. The 'G' portion of the name will be replaced by the letters 'B' through 'Z' in the event more than one table occupies the same tablespace. The actual name will be maintained in a Production cross reference DB2 table DBMADM.CROSS_REFERENCE).
2.   All objects which are referenced in SELECT, UPDATE, INSERT, and DELETE operations will be 'Declared' in the program.
3.   Explicit column identification will be used for all SELECT, UPDATE, and INSERT operations.

TEMP DATABASES AND TABLESPACES

As of DB2 V9, all TEMP database objects have moved to the Workfile database (DSNDB07 & ssidWRK7). Therefore, the tablespace naming guidelines will follow the SYSTEM DBA's guidelines.

A WORKFILE database cannot be shared between DB2 subsystems or data sharing members. PUBLIC implicitly receives the CREATETAB privilege (without GRANT authority) to define a declared/create global temporary table in the WORKFILE database. In Test, Agency Adm and Project DBA groups (and in some cases PUBLIC in DBT) has USE privilege to use the Storage Group and BP (for tablespace/index) in the WORKFILE database. In Production, the Agency Adm and Project DBA groups have the USE privilege to use the Storage Group and BP (for tablespace/index) in the WORKFILE database. These explicit privileges are recorded in the DB2 catalog.

STORED PROCEDURES

Use of the CCF/MMF Clist **%CCFR9001** to generate stored procedures should be encouraged. The ZQ id should be the OWNER of these procedures to standardize ownership. These guidelines for developing these objects are subject to changes based on shared experience of the DBA's and new releases of db2.

NAMING GUIDELINES: **aaaSSnnn**

Where:	**aaa**	system acronym (e.g. CSE for CHARTS)
	SS	constant to denote a Stored Procedure.
		SP – external Stored Procedure.
		NS – Native SQL Procedure (introduced in DB2 V9).
	nnn	three alphanumeric characters

TRIGGERS

These guidelines for developing these objects are subject to changes based on shared experience of the Project DBA's. The creation and support of DB2 TRIGGERS will be a DBA FUNCTION and will be the responsibility of the Project DBA's.

NAMING GUIDELINES**: aaaTRnnn**

Where:	**aaa**	system acronym (e.g. NFO for NFOCUS, etc.)
	TR	constant to denote a trigger
	nnn	three alphanumeric characters

USER DEFINED FUNCTIONS (UDF)

These guidelines for developing these objects are subject to changes based on shared experience of the Project DBA's. The creation and support of DB2 UDF's will be a Project DBA FUNCTION and will be the responsibility of the Project DBA's. There are different types (e.g. scalar, column, and table) of functions so use SQL Reference manual for guidelines.

NAMING GUIDELINES**: *aaaaaaaaa***

Open format. Use meaningful names (e.g. CALC_BONUS) that this new function represents. The name must not be any system-reserved keywords.

Reference Guide
LOBS (Large Objects)

NAMING GUIDELINES:

LOB tablespace **NFOTAnnn**   where A indicates this contains an Auxiliary table and nnn is the same value as the tablespace that has the table with the defined LOB column.

Index for the Aux table **NFOAnnn1**   same rules as for the above index names except we have added the A to indicate that this is an index for an Auxiliary table.

Auxiliary table names t**ablename_AUX** same name as table name to which it is paired plus AUX.

We would really like to keep the character representing the auxiliary index to one character so it would still conform to the original naming standard (it doesn't have to be "A") - long-term, the single character

would still identify the object as an aux index, but more importantly it will should make the implementation of the SMS ACS routines easier to administer.
There is one other issue. If there is more than one LOB column defined in a table, it means we will end up with more than one LOB tablespace, Auxiliary table and index for the base table. The index name defined above can handle this since the last character of the name is a sequence number. For the Auxiliary table name we will add a sequence number to the end of the table name, such as tablename_AUX_1. The tablespace is a problem though. The only portion of the name we could use is the TA portion on NFOTAnnn. Some possibilities are; T!..T2..

OBJECT OWNERS AND CREATORS

Objective:

The System DBA will create all DB2 databases, all DB2 objects within production system and development system data bases (version 00 - 19 numbered data bases in the development tiers), and will grant appropriate access to objects within those data bases to other selected RACF Groups/ids.

RACF will be used to create secondary authorization ids. It will also be used to control underlying data set access.

All DB2 tables, created by Project DBA, will use an Agency Adm. Group id as the 'creator' of the tables. This Agency Adm. group id will normally use an agency acronym for the first three positions of the id. For large projects, a project acronym may be used in place of the agency acronym. This full qualifier (as described below) will be used in the bind process to designate the qualifier of the tables referenced in the program or package. This qualifier will use the following naming convention:

NAMING GUIDELINES: **xxxADtn**

Where: **xxx**   = Agency or Project Acronym

AD = Literal for Adm.
t = Tier

        'U' = Unit
        'S' = String
        'C' = CAT
        'T' = Training
        'M' = Production

n = version or sequence letter or number *

* A version is only used in the Applications Development and Training environments. Versions will not be used in the production environment.

Example:

DSSADS1 = Department of Social Services creator at the String Level, Chain A (Path 1)
DASADM = Department of Administrative Services in the Production Level

1. System DBA, or Project DBA, or the appropriate system RACF administrator, will establish secondary authorization ids for every system, subsystem, project, or major project cell.

2. Authorization ID's will be created using the following naming convention:

Secondary Authorization Id = **xxxuuts**

Where:

xxx = Department (System) Acronym
uu = Type

        'AD' = DBADM authority
        'CL' = DBCTL authority
        'AP' = Application
        'US' = Client or User
        'WA' = WLM Applications (any additional WLM application racf groups will be WB-WZ with WB being the next one)

t = Tier

        'U' = Unit
        'S' = String
        'C' = CAT
        'T' = Training
        'D' = Development
        'P' = Production

    s   = Test Path Designation Number*

*A number, or a letter, designating the testing 'Path' (e.g. '1', '2' or '3') will be used after the Tier designation. The number '0' will be used to designate access to all objects under the high level System acronym for application or task specific groups. The letter 'T' will be used with the xxxADM designation to indicate project DBA authority in the entire test environment for that system.

In production this number will be used to separate differing levels of authorization. Generally, a zero ('0') will indicate all authorities. A '1' or above will indicate specific authorities.

Individual TSO ids may be substituted for group ids for specific authorities.

*NOTE   See Appendix A for an example of Owner and Qualifier names

    The System DBA will be the owner of the AD and CL RACF groups. The Application areas 'can own' the AP, US and WA RACF groups. The secondary authorization id groups will generally be assigned capabilities or accesses in the following areas:

xxxADts =   all DBADM authorities as outlined in the Current DB2 Administration Guide. The System DBA/Project DBA will normally be the only people placed into these groups.

xxxCLts =   DBCTL data base capabilities without the DROP, CREATETS, and CREATETAB authorities. People placed in these groups will normally be responsible for the maintenance of the DB2 tables in the development environment.

xxxAPts =   all access to table data including SELECT, UPDATE, INSERT, and DELETE. This group will generally contain the applications developers who are responsible for the development or maintenance of the programs within the system. These secondary authorization groups will generally have ownership of packages (i.e. programs) in the development environment.

xxxUSts =   SELECT authority only for the tables in a system. This group will generally contain the clients or users of the system who require ad-hoc READ access to the tables and data in the system.

xxxWAts =   EXECUTE authority on stored procedures or functions. This group will generally contain the analysts/users of the system to be able to call stored procedures/functions.   If additional racf groups are needed for the applications the WA will be WB-WZ with WB being the second racf group after WA.

3. System DBA or Project DBA will assign project staff to the secondary authorization groups in the production DB2 subsystems as required.

   Agency or system RACF administrators will assign project staff to the secondary authorization groups in the development subsystems for table authority.

   Project Managers or designates will be responsible for notifying System DBA when an individual is to be added or removed from a production subsystem secondary authorization group.

4. Project DBA will provide master or sample DDL to create objects at the unit level.

5. Project DBA will be responsible for maintaining the DB2 objects in all test environments and Production environments.

   Exceptions to this provision may be approved by the Unit Supervisor to System DBA, when compelling reasons for those exceptions are warranted.

6. Project DBA will do all DCLGEN's for production system and development system tables and will place the resultant copybooks or include members in the appropriate libraries.

7. Individuals will be responsible for all DB2 objects, which are defined, or developed in their personal version data bases. This responsibility includes table definitions and individual DCLGEN's.

8. System DBA/Project DBA will hold selected SYSCTRL authorizations, PACKADM, and DBADM control of all DB2 objects.

9. Except for normal maintenance capabilities possessed by the Project DBA group, Client Agencies will maintain ownership over all data stored in DB2 objects and will have final approval regarding access to that data.

PLANS AND PACKAGES

1. System DBA will be responsible for creating all PLAN's in both the test and production environments. Authorizations will then be granted to the appropriate Secondary Authorization ID.

2. In the Test DB2 environment, the following guidelines apply:

    a. All plans will be created with an OWNER of DBMADM. DBMADM then grants PACKADM on each collection to the appropriate level Authorization ID for the system, level and path (e.g. DBMADM grants PACKADM on collection RFSBATS1 to DSSADS1). This authorization provides CREATE, BIND and EXECUTE for all packages in the collection using an owner of the Secondary Authorization ID for the system, level and path.

    b. The QUALIFIER of all plans and packages will be the Agency Admin. Secondary Authorization ID for the system, level and path (e.g. DSSADS1 would be the qualifier for the RFS plans in the String, Chain A (Path 1) environment).

    c. At the UNIT level, the QUALIFIER can be overridden to allow a program to access an individual's test tables. Example: using CCF/MMF, type 'DB2' on the command line of the CCF/MMF screen. This will display a DB2 override panel. Enter the owner of the tables you wish to access in the QUALIFIER field.

        NOTE: In order to do this, the following conditions must be met:

        i   A table, alias or synonym for EVERY table accessed in the program must exist with a CREATOR equal to the QUALIFIER (e.g. if the program accesses TABLE_A and TABLE_B, and TABLE_B is owned by USERID, USERID must create a synonym called TABLE_A that points at the version of TABLE_A that is to be referenced).

        ii   Access to the tables owned by the ID specified in the QUALIFIER must be granted to PUBLIC (e.g. if the program is going to perform SELECTs and INSERTs on USERID.TABLE_A, then USERID must grant SELECT and INSERT on TABLE_A to PUBLIC).

3. In the Production DB2 Environment, the following guidelines apply:

    a. The OWNER and QUALIFIER of all plans and packages in production will be the appropriate Agency Secondary Authorization ID for the system. e.g. DASADM is the owner and qualifier of the CCF production plans. NOTE: Before an Agency Secondary Authorization ID can bind a plan using Package Lists, it should be granted PACKADM on the system collections and EXECUTE on the DBM collections.

4. The plan name will normally be 8 bytes in length.

    a.   The first three bytes will be the system acronym
    b.   The second three bytes will be the environment
    c.   The last two bytes of the plan name will correspond to the tier in which the plan is executed. The following naming convention will be used:

        NAMING GUIDELINES for Application Plan Name**: xxxeeets**

        where:

                xxx = Department (System) Acronym
                eee = Environmental designation
                      'ONL' = Online
                      'BAT' = Batch

                t = Tier designation
                      'U' = Unit
                      'S' = String
                      'C' = CAT
                      'T' = Training
                      'A' = Assist GT
                      'P' = Production

                s = Test Path Designation Number (0 for production)

5. A program will be bound as a package and will normally be assigned to a collection-id, which has the same System Acronym. The mid-portion of the collection-id will be determined in the following manner:

    a.   An environment collection-id (e.g. XXXONLts, XXXBATts, XXXOBJts) when a module is only used in one environment (i.e. this name is normally the plan name under which the package is bound).

    b.   A combination batch and online collection-id (e.g. XXXCOMts) when a module is used by both the batch and online in the same system.

    c.   A System DBA collection-id (e.g. DBMBATts, DBMONLts or DBMCOMts) when a module is used in two or more systems, and where the module performs one or more universal functions which are not the specific function of a system process (e.g. date conversion routine, etc).

6. All Collection IDs will use the following naming convention:

    NAMING GUIDELINES Application Collection-id: **xxxeeets**

    where:

    xxx = Department (System) Acronym or 'DBM'

    eee = Environment (Online, Batch, or Combination)
           ONL = used only in Online programs
           BAT = used only in Batch programs
           OBJ = used only in Object programs (i.e. stored procedures/triggers)
           COM = Combination Online and Batch

    t = Tier designation
           'U' = Unit
           'S' = String
           'C' = Client Acceptance Test
           'T' = Training
           'A' = Assist GT
           'P' = Production

    s = Test Path Designation Number (0 for production)

7. PLAN TABLE(s):

Plan tables, used by applications personnel for explaining plans/packages, are SOLELY the responsibility of the application user. If the data is to be saved/protected, that person (or the respective Project DBA) must set up their own tablespace/table.

OBJECT CREATION AND AUTHORIZATIONS

The following objectives should be followed when creating objects and granting authorizations.

1. Objects should not be created with an individual ID as the CREATOR or OWNER. The CREATOR should be a RACF Agency Adm. Group ID. The recommended CREATOR of each object type will be identified in the following sections.

2. Grants should not be made from individual IDs, but instead should be made from a RACF group. Those rare grants that require SYSADM authority should be made from the DBMZ001 ID. This will allow any of these special grants and revokes to be controlled by the DBM group.

3. Wherever possible, grants should be made with the CURRENT SQLID set to DBMADM. This should minimize the need to constantly change the CURRENT SQLID. Because some objects are not owned by DBMADM, there are cases where DBMADM does not have the authority to perform the grant. In these cases, the authority should be granted from the owner to DBMADM with grant, and then subsequent authorizations should be granted from DBMADM. The instances where this is necessary will be identified in the following sections.

4. The grants identified in this document are the minimum, or default, authorization levels. Additional grants may be made, but generally only with an approved DB2 RACF request. Where appropriate, these optional authorizations will also be documented.

5. Exceptional or temporary grants should be made to individual IDs. Examples of this include the database authorizations that are permanently required by CDPPROD to run scheduled utilities, and temporary authorities such as LOAD on a production database or EXECUTE on a production batch plan.

6. SYSADM authority should be used to grant only where documented. The only grants that have been identified that must be done with SYSADM authority are the initial system authorizations that are granted to DBMADM. If additional authorizations must be granted using SYSADM authority, they should be documented here. This point is not in reference to the RACF group named SYSADM.

7. The old RACF group that is named SYSADM should not be used to perform any new grants. This RACF group generates confusion since it is named the same as a DB2 authorization level that it no longer holds. There are still a number of objects that have SYSADM as the creator. In these cases, authority on the object should be granted from SYSADM to DBMADM with grant, and then subsequent grants can be made from DBMADM.

8. Whenever possible, while objects are being maintained or migrated, if the creators of objects or grantor of authorizations are different than the recommendations documented here, replace them.

9. On the surface, these recommendations may appear to be merely replacing SYSADM as the owner of objects and grantor of authorizations with DBMADM. There are several reasons for the change that explain why this is not the case.

10. Previously, SYSADM was the owner of objects and grantor of most authorizations since this RACF group held SYSADM authority. There is no need to use SYSADM authority on a daily basis if the proper grants are made. In fact, it could be rather dangerous to constantly have our SQLID set to a RACF group that has SYSADM authority. The reason DBMADM has been chosen as the source of most of the grants is to prevent the confusion of having a RACF group named the same as a DB2 system authorization. At some point in the future, the SYSADM RACF group can probably be eliminated.

11. Having SYSADM as the creator of all of the tables meant that a given table name could only be used once in the entire system. Because of the number of agencies that we support, and their independence from each other, it is preferable to treat each agency as a separate enterprise. By using an agency level qualifier, agencies no longer need to worry about colliding with a table name used by another agency.

12. Some agencies have the need to perform their own production binds. Previously, this would require that these agencies hold the authorities to create and own the plans, collections and packages. In many cases, we would then not have sufficient authority to rebind their plans and packages when performing maintenance (without using SYSADM). By using the agency level owners and bind agent authority, we are able to allow these agencies to run their own binds and still maintain control of the plans, collections and packages.

13. Project DBAs can request and obtain specially authorized, no logon, IDs that they can use only to run their utilities and maintenance jobs. Since these IDs have typically restricted access, they should be guarded and only used for their specified use.

SYSTEM AUTHORIZATIONS

There should be two IDs identified as Installation SYSADM on each subsystem. Currently, these IDs are DBMZ001 and, in the case of PR, PR1ADM or PR2ADM. The DBMZ001 ID should be used to grant all necessary authorities to DBMADM(System DBA), and then subsequent grants should originate from there.

The following authorities should be granted from DBMZ001 to DBMADM on all subsystems:

- SYSCTRL WITH GRANT OPTION
- PACKADM ON COLLECTION * WITH GRANT OPTION

On the Development and Production subsystems, BINDAGENT authority is used to allow the binds of plans and packages without changes in ownership.

On the Development subsystem, each Agency Adm group (i.e. **aaaADtp**) should grant BINDAGENT TO PUBLIC to authorize binds at all test levels.

On the Production subsystem, each Agency Adm group (i.e **aaaADM)** should grant BINDAGENT TO DBMOPER, ZQGTCDX (in DB2) or to PUBLIC (in DB3, DB4, DB5).

BUFFERPOOL AUTHORIZATION

The SYSCTRL authority held by DBMADM(System DBA) includes USE on all defined bufferpools. Therefore, as long as DBMADM authority is used when creating tablespaces and indexes, no additional grants are required for us to maintain these objects in any bufferpool.

However, since Project DBA's can create these objects under certain circumstances, USE OF BUFFERPOOL should be granted from System DBA to Project DBA RACF Groups. Granting to PUBLIC should be limited and/or eventually removed.

STOGROUP CREATION AND AUTHORIZATION

All STOGROUPs on all subsystems should be created by DBMADM(System DBA). As the CREATOR of the STOGROUP, DBMADM implicitly holds all authority on that STOGROUP with grant ability. For old STOGROUPs that have a CREATOR other than DBMADM, the STOGROUP creator should grant USE on the STOGROUP to DBMADM WITH GRANT OPTION. Subsequent grants can then be made from DBMADM.

DATABASE CREATION AND AUTHORIZATION

All databases should be created by DBMADM(System DBA). As the CREATOR of the database, DBMADM implicitly holds all authority on that database with grant ability. For legacy databases that have a creator other than DBMADM, the database creator should grant DBADM TO DBMADM WITH GRANT OPTION. Subsequent grants can then be made from DBMADM.

For new databases the Project DBA's will provide the System DBA's:

1. The new Database name.
2. The (default) Storage group to USE.
3. The (default) BP's to USE for tablespaces and indexes.
4. The new Online and/or Batch plan names and Qualifier (if known).

The System DBA will grant DBADM WITH GRANT db authority to Project DBA Secondary RACF ADM Group; along with execute privilege to the appropriate groups.

1. **INDIVIDUAL DATABASES** - (sssUDBxx) where xx is the individual's initials.
   When databases are created for individuals to use, DBMADM should grant DBADM WITH GRANT to the individual's MVS ID. This will allow the individual to perform all operations within the database, and grant authority to whomever else may need it.

2. **UNIT LEVEL DATABASES** - (sssUDB0p)
   For unit level databases, DBMADM should grant DISPLAYDB, IMAGCOPY, LOAD, RECOVERDB, REORG, REPAIR, STARTDB, STOPDB, STATS to the CL RACF group for that system, tier and path (sssCLtp).

3. **STRING AND CAT LEVEL DATABASES** - (ssstDB0p)
   At the string and CAT levels, DBMADM should grant DISPLAYDB, IMAGCOPY, LOAD, RECOVERDB, REORG, REPAIR, STARTDB, STOPDB, STATS, TO CDPPROD and to the CL RACF group for that system, tier and path. (sssCLtp)

4. **TRAIN AND ASSIST LEVEL DATABASES** - (ssstDB0p)
   At the Training and Assist/GT levels, DBMADM should grant DISPLAYDB, IMAGCOPY, LOAD, RECOVERDB, REORG, REPAIR, STARTDB, STOPDB, STATS, TO CDPPROD and to the AWPCL group for the tier and path. (AWPCLtp)

5. **PRODUCTION LEVEL DATABASES** - (sssPDBIx)
   At the production level, DBMADM should grant DISPLAYDB, IMAGCOPY, REORG, REPAIR, STARTDB, STOPDB and STATS to CDPPROD. Additional authorities on production databases can be requested via the DB2RACF form, and should be granted from DBMADM to the individual's IDs.

TABLESPACE CREATION AND AUTHORIZATION

All tablespaces controlled by the System DBA group should be created by DBMADM. All tablespaces controlled by the Project DBA's should be created by Project DBA Secondary RACF ADM group.

TABLE CREATION AND AUTHORIZATION

All tables should be created using DBADM authority, but with an agency level qualifier. As the creator of the database, Project DBA ADM groups hold the necessary CREATE table authority. Using an agency level qualifier allows each agency to operate as a separate enterprise and name their tables without regard to table names used in other agencies. In production, an agency qualifier in the form aaaADM should be used. The agency level qualifier of all test tables should also include the tier and path designation: aaaADtp.

1.   **UNIT LEVEL TABLES** - (aaaADUp.table_name)

     At the unit level, Project DBA ADM groups should grant SELECT, INSERT, DELETE and UPDATE on all tables to its associated RACF group, xxxAPtp.

2.   **STRING, CAT, TRAIN and ASSIST/GT LEVEL TABLES** - (aaaADtp.table_name)

     At each of these levels, Project DBA ADM groups should grant SELECT, INSERT, DELETE and UPDATE on each database's tables to its associated applications RACF group, sssAPtp.

3.   **PRODUCTION LEVEL TABLES** - (aaaADM.table_name)

     At the production level, Project DBA ADM groups should grant SELECT, INSERT, DELETE and UPDATE on each database's tables to its associated sssAPP0 RACF group. DBMADM (System DBA) or Project DBA ADM groups should also grant SELECT to the sssAPP1 and sssUSP0 RACF groups. Users requiring "standard" access to the production tables will be placed into one of these existing RACF groups upon receipt of an approved DB2RACF request form.

               sssAPP0 - Applications personnel requiring update capability.
               sssAPP1 - Applications personnel requiring read-only access.
               sssUSP0 - End-users requiring read-only access.

     Users requiring "non-standard" access to the production tables (i.e. end-user update or access to selected tables only) may be placed in a special RACF group or may have the authority granted directly to their ID. Any of these grants should originate from the DBMADM (System DBA) RACF group.

INDEX CREATION AND AUTHORIZATION

Indexes should be created in the same way as tables, using DBADM authority and agency level qualifiers. No additional authorizations are required.

ALIAS CREATION AND AUTHORIZATION

When used, Aliases should be created using <u>DBADM</u> authority and an agency level qualifier. No additional authorizations are required.

VIEW CREATION AND AUTHORIZATION

Views should be created in the same way as tables, using DBADM authority and agency level qualifiers. Immediately after creation, only the creator of a view holds authority on it.

Therefore, the agency level RACF group (aaaADM or aaaADtp) must grant SELECT on the view to DBMADM(System DBA) with grant option. Then, all subsequent grants can be made using DBMADM authority.

5. **UNIT LEVEL VIEWS** - (i.e. aaaADUp view_name)
   At the unit level, DBMADM should grant SELECT on all views to its associated applications RACF Group, sssAPtp.

6. **STRING, CAT, TRAIN and ASSIST/GT LEVEL VIEWS** - (i.e. aaaADtp view_name)
   At each of these levels, DBMADM should grant SELECT on each database's views to its associated applications RACF group, sssAPtp.

7. **PRODUCTION LEVEL VIEWS** - (i.e. aaaADM.view name)
   At the production level, DBMADM should grant SELECT on each database's views to its associated sssAPP0, sssAPP1 and sssUSP0 RACF groups. See the PRODUCTION LEVEL TABLES section for an explanation of these RACF groups, and rules for additional authorizations.

System DBA's will be responsible for authorizing all appropriate Secondary Auth ids.

1. **PLAN CREATION AND AUTHORIZATION**
   With the exception of applications that use dynamic plan allocation, all plans on all
   subsystems should be created with an OWNER of DBMADM(System DBA).
   This allows us to grant and revoke all plan authorities from this RACF group.
   Previously, when plans were owned by the agency level group, users who had
   been granted BIND AGENT authority were able to rebind the plans with
   different parameters, and there is not a need for this ability.

2**.** **TEST PLANS** - (sssONLtp/sssBATtp)
   EXECUTE on all test (UNIT, STRING, CAT, TRAIN & ASSIST) plans should be
   granted from DBMADM to PUBLIC.

1. **PRODUCTION BATCH PLANS** - (sssBATP0)
   EXECUTE on all production batch plans should be granted from DBMADM to
   appropriate Production ID used to execute scheduled jobs (e.g.-CDPPROD, LABPROD,
   or DOR9999). EXECUTE authority on production batch plans can be requested via the
   DB2RACF form, and should be granted from DBMADM to the individual ID. This is
   generally a temporary authorization that should be revoked after a pre-determined
   number of days.

2. **PRODUCTION ONLINE PLANS** - (sssONLP0)
   EXECUTE on all production online plans should be granted from DBMADM to
   PUBLIC, since we will rely on CICS security to prevent unauthorized access. Because of
   this, all production online plans should ENABLE (CICS) to prevent access from any
   other attachment.

3. **PRODUCTION ONLINE PLANS - DYNAMIC PLAN ALLOCATION**
   Plans that are used via dynamic plan allocation are still DBRM binds and need to be
   bound with an aaaADM owner. This will enable those individuals granted BIND
   AGENT from the aaaADM to bind these plans. The aaaADM RACF group that owns
   these plans should grant EXECUTE on the plans with grant option to DBMADM. Then,
   the grant of EXECUTE to PUBLIC can be made from DBMADM.

4. The Agency Secondary Authorization ID that owns the plan can optionally grant BIND
   AGENT to another Primary or Secondary Authorization ID.   This authorization provides
   CREATE and BIND for all packages in collections on which the Agency Secondary
   Authorization ID holds PACKADM authority.

Collections are not explicitly created they exist as a logical foreign key on the SYSPACKLIST and SYSPACKAGE catalog tables. Before a plan or package can be bound using the collection, however, the appropriate authorizations must be in place. Even with SYSCTRL authority, DBMADM does not have the authority to bind a plan using a package list, or bind a package. The SYSADM (DBMZ001) id must first grant DBMADM PACKADM authority on all collections (*) with grant option. In conjunction with the BIND ADD authority that is part of SYSCTRL, this will enable DBMADM(System DBA) to bind plans using package lists.

1. **UNIT, STRING, CAT, TRAIN, AND ASSIST/GT LEVEL COLLECTIONS** - (sssONLtp, sssBATtp, or sssOBJtp)
   At each testing level, DBMADM should grant PACKADM on each collection to its associated agency RACF group (aaaADtp). In conjunction with the grant of BIND AGENT from each of the agency RACF groups to PUBLIC, this will allow anyone to bind packages at all test levels.

2. **PRODUCTION LEVEL COLLECTIONS** - (sssONLP0/sssBATP0/sssOBJP0)
   At this level, DBMADM should grant PACKADM on each collection to its associated production agency RACF group (aaaADM). In conjunction with and grants of BIND AGENT done from each agency RACF group, this will provide the ability to bind production packages, but only within the agency for which BIND AGENT was granted.

PACKAGE CREATION AND AUTHORIZATION -
(sssONLtp/sssBATtp.package)

There are no additional authorizations required at the package level. The authority to bind and execute packages is controlled at the collection and, indirectly, at the plan level. This section will address what OWNER and QUALIFIER should be used for packages at each level.

1. **UNT, STG, CAT, TRAIN and ASSIST/GT LEVEL PACKAGES**
   (sssONLtp/sssBATtp/sssOBJtp pkg)

   Packages at all testing levels should be bound with and agency level OWNER and QUALIFIER that includes the tier and path (aaaADtp).

2. **PRODUCTION PACKAGES** - (sssONLP0/sssBATP0/sssOBJP0.package)

   Production packages should be bound with an agency level OWNER and QUALIFIER (aaaADM).

Reference Guide
AUTHORIZATIONS REQUIRED FOR OBJECT PROMOTION

Objective:

Project DBA will promote database objects, plans and packages only with the approval of the appropriate project person.

1. DB2 objects, plans and packages will be promoted to the next level of test or production only with the signature of the responsible person as listed below:

    a.   String   = Project Analyst
    b.   CAT   = Lead Analyst
    c.   Training   = Lead Analyst
    d.   Production   = Project Manager

2. The next higher authority may sign off objects for a lower authority. A lower authority can only be assigned to provide signatures if approved by a higher authority than the person for whom the provision is being made (e.g. The Project Manager may approve a Project Analyst signing plans into CAT in the absence of the Lead Analyst).

3. Production plan or package binds may be accomplished by an On-call analyst or programmer, but only to replace a plan or package currently in production. These actions require the signature of the Project Manager by the close of the next business day. In these instances, the signature insures that the manager is aware of the emergency and the actions taken to remedy the production problem. Failure to secure this or another approved signature will result in the plan or package being freed.

PRIORITY and TIMELINESS OF OBJECT PROMOTIONS

Objective:

Project DBA will provide data base DML, DDL and DCL support in a timely manner.

1. Proper and adequate planning contributes to the successful implementation of database objects or programs. To this end, sufficient prior notification of schedules to move objects or programs must be given. Your poor planning in not my emergency!

2. The following matrix provides a reasonable timeframe for the movement of objects through the test levels:

    a.   Plan or Package bind = 1 hour per program
    b.   Table alter/creation   = 1 hour per object
    c.   Grants or Alias create = 1 hour per object

3. The following matrix provides a reasonable timeframe for the movement of objects to production after notification:

    a.   Plan or Package bind = 1 hour per program
    b.   Table alter/creation   = 1 hour per object
    c.   Grant or Alias create = 1 hour per object

4. These changes can be expected to be made at the rate of 1 movement per hour. The project leader should tally all movements and compute the lead-time necessary to move these objects through test and production. In those cases where other system changes must be made, or where changes must await system disconnects and reloads, these changes or moves may be delayed until the next cycle unit.

5. Unless written notice and explanation is provided, all Table additions or changes will be accomplished WITHOUT data recovery. Sufficient notice and analyst cooperation will be needed to accomplish changes with data recovery.

USE OF AUTHORIZED TOOLS FOR OBJECT PROMOTION

Objective:

System DBA/Project DBA will use only authorized software tools or products to promote database objects.

1. Promotions or the movement of objects through the test and production tiers will be accomplished using only authorized Data Base Management software tools or products.

USE OF AUTHORIZED SOFTWARE FOR PROGRAM PROMOTIONS AND BINDS

Objective:

A common methodology and software will be used to promote programs and accomplish DB2 binds, at all levels of development and into production.

1.   Promotions or the movement of objects through the test and production tiers will be accomplished using the CCF/MMF Migration methodology and software. See Appendix C for a description of the promotion strategies to be used for normal and emergency migrations.

TRANSITION

Objective:

A Project DBA staff person will ensure that all database objects, and appropriate copy and recovery procedures are in a receiving environment during the transition and promotion phases of database objects.

1.   System DBA/Project DBA will be responsible for all maintenance and recovery procedures for DB2 objects in the production environment (see Document DB2 Data Base Management Systems Procedures).

2.   The CCF staff will be responsible for the migration of normal Application Development programs and associated objects through the CCF chains.

3.   All conversion and/or one-time programs must be successfully executed in a String environment. Project DBA must approve the results of these programs before they may be promoted and run in the production environment.

4.   The DB2 Load Utility will be used for large or mass inserts of information into DB2 tables. Those jobs will use the 'LOG NO' parameter (i.e. do not log the inserts) for those inserts. System DBA/Project DBA may authorize exceptions to this procedure when processing requirements justify that exception. Running an Image Copy after this is advisable.

5.   All Migrations and Conversions of DB2 objects will use a formal and written script to plan, implement, and verify the movement or change of DB2 objects in the production environment. Appendix D, DB2 Migration and Conversion Procedures Checklist, may be used as a skeleton script. Alterations, revisions, additions, and omissions should be made based upon the unique requirements of each project.

PRODUCTION

Objective:

System DBA/Project DBA will ensure that all access to data base objects, and information contained in those objects is done only for the business of the State, is done in a manner which ensures the integrity of the data base objects, and the data contained in those objects, and is accessed in the most efficient and secure manner possible.

1. Project DBA will direct or accomplish timely maintenance of DB2 tablespaces and their indexes. This maintenance includes:

   - Regular image copies of all tablespaces.
   - Resizing of tablespaces or index spaces as required.
   - Reorganization of tablespaces to effect changes to those data sets.

2. DB2 objects, including tablespaces, tables, and indexes will only be recovered using image copies and/or system logs when those objects have been rendered unusable by a System or DASD device failure, and when no other means to recover the object is available. These extraordinary recovery procedures will not be used to reverse the alteration of data within those objects, unless that recovery is directed and approved by director(s) of the responsible business areas and the senior manager(s) of CDP.

CHANGING PRODUCTION TABLES and PROGRAMS

Objective:

Production tables will normally be changed using the DB2 ALTER option rather than using the DROP option.

1. A table which is defined in production, and which contains viable business data, will normally not be deleted (DROPPED) and redefined to effect changes to column name or order. In situations where it is felt that changes in column names or orders must be made, the creation and use of a VIEW is preferred.

2. In situations where changes to column names and column order cannot be accommodated with the use of a VIEW, or when the data type of a column must be changed and the table must be DROPPED and CREATED, the following responsibilities will be observed:

   a. The responsible application development area will schedule an outage with all affected parties, including all user, technical support, and operations staff.

b. The responsible application development area will unload the table data using either the standard DSNTIAUL unload program, or an equivalent program.

c. Project DBA will DROP the old table and CREATE the new table based upon the information furnished by the responsible application area.

d. The responsible application development area will convert the old data to the new format.

e. The responsible application development area will load the converted data in its new format.

f. Project DBA will arrange for standard Image Copy and, if necessary, CHECK DATA utilities to be run.

g. The responsible application development area will examine the output from the CHECK DATA utility and resolve all rejected data or consistency conflicts.

h. The responsible application development area will examine the data for accuracy and consistency.

i. Project DBA will arrange for standard RUNSTAT and BIND utilities to be run.

j. Project DBA will remove all prior references to Image Copy data sets from the SYSIBM.SYSCOPY table; which were made prior to the DROP table action.

k. The responsible application development area will insure that the new table, the effected programs, QMF queries, and procedures are operational.

3. In situations where changes to tables and programs will be made through the DROP and CREATE of a table, Project DBA will refer to the checklists found in Appendix D, "DB2 MIGRATION/CONVERSION CHECKLIST" to guide and suggest the actions which should be followed in the conversion process.

POST IMPLEMENTATION REVIEW

Objective:

Project DBA will conduct a Post Implementation Review for all new systems or for all projects; which significantly changed a current system.

1. A Post implementation review will be held three (3) to six (6) months after a project has gone into production.
2. Participants in this review will include staff involved in the design, construction, implementation, and operation of the new or changed system.
3. This review will cover, but not be limited to, the following topics:
    a. Recommended changes to Standards, Procedures, and/or Guidelines
    b. System strengths and deficiencies
    c. Development software and tool strengths and deficiencies
    d. Plans to correct design and/or performance problems

RESTART TABLES

Restart tables are used to facilitate checkpoint restart, using the CHECKPOINT RESTART module ADMCOM01. Two tables are used RESTART_DFLT, and RESTART_CNTL.
Batch programs will take commits at regular pre-defined intervals, so that if an ABEND occurs, a minimum amount of re-processing will be required.

If the program/job has not previously been registered for checkpoint restart, the program will call the ADMCOM01 module, and obtain (READ) default control values (commit frequency) from the RESTART_DFLT table, based on the system acronym.

A RESTART_CNTL entry is then created (INSERT), for the program/job, with commit frequency obtained from RESTART_DFLT, and blank restart data.

DB2 timestamp is obtained, and saved as the last commit timestamp, to determine when the next commit/checkpoint is to be taken.

Commits are taken after the appropriate interval has been reached.
The RESTART_CNTL is updated with DB2 timestamp as the last commit timestamp, and the restart data is updated.

Restart is based on the restart data found in RESTART_CNTL.

At normal completion of the program/job, the restart data is cleared.

APPENDIX A

The following example uses the Medicaid system, which has a system acronym 'MCP' and an Agency acronym of 'DSS'.

Medicaid Table

	APPL-ID (OWNER)	DATABASE NAME	TABLESPACE NAME	QUALIFIER	TABLE NAME
UNIT	DSSADU(1-9)	MCPUDB0(1-9)	MCPTS001	DSSADU(1-9)	Any name
STRING	DSSADS(1-9)	MCPSDB0(1-9)	MCPTS001	DSSADS(1-9)	Any name
CAT	DSSADC(1-9)	MCPCDB0(1-9)	MCPTS001	DSSADC(1-9)	Any name
PROD	DSSADM	MCPPDBII	MCPTS001	DSSADM	Any name

1. The Owner of all packages bound at all test environments would be DSSADts as executed by the CCF group using their BIND AGENT authority.

2. The Qualifier for all test tables/indexes/views would be 'DSSADts'.

3. The Qualifier for all Production tables would be 'DSSADM'.

APPENDIX B

CICS Regions and DB2: OTHER SYSTEMS DEVELOPMENT CHAINS

	Chain A/Path 1	**Chain D/Path 2**	**Chain B/Path 3**
UNIT	TESTCEUA TESTCEWA TESTCEPZ	TESTCEUD TESTCEUP – DBX	TESTCEUB
	DB = sssUDB01	DB = sssUDB02	DB = sssUDB03
	QUAL = aaaADU1	QUAL= aaaADU2	QUAL = aaaADU3
	GRP = sssAPU1	GRP = sssAPU2	GRP = sssAPU3
	PLAN = sssONLU1	PLAN= sssONLU2	PLAN = sssONLU3

STRING	TESTCESA TESTCEAA	TESTCESD	TESTCESB TESTCESP - DBX
	DB = sssSDB01	DB = sss**S**DB02	DB = sssSDB03
	QUAL = aaaADS1	QUAL= aaaAD**S**2	QUAL = aaaADS3
	GRP = sssAPS1	GRP = sssAP**S**2	GRP = sssAPS3
	PLAN = sssONLS1	PLAN= sssONL**S**2	PLAN = sssONLS3

CAT	TESTCECA	TESTCECD TESTCECP - DBX	TESTCECB
	DB = sssCDB01	DB = sss**C**DB02	DB = sssCDB03
	QUAL = aaaADC1	QUAL= aaaAD**C**2	QUAL = aaaADC3
	GRP = sssAPC1	GRP = sssAP**C**2	GRP = sssAPC3
	PLAN = sssONLC1	PLAN= sssONL**C**2	PLAN = sssONLC3

Train	TESTCETA TESTCETE		TESTCETB
	DB = sssTDB01		DB = sssTDB03
	QUAL = aaaADT1		QUAL = aaaADT3
	GRP = sssAPT1		GRP = sssAPT3
	PLAN = sssONLT1		PLAN = sssONLT3

ASSIST GT	TESTCEAA		
	DB = sss**A**DB01		DB = sss**A**DB03
	QUAL = aaaAD**A**1		QUAL = aaaAD**A**3
	GRP = sssAP**A**1		GRP = sssAP**A**3
	PLAN = sssONL**A**1		PLAN = sssONL**A**3

TESTCE** are DBT unless noted

sss   = System Acronym   (e.g. MCP, PAE, NAS, etc.)

aaa   = Department Acronym   (e.g. DSS, NSP, DAS, etc.)

N-FOCUS/MEDICAID DEVELOPMENT CHAINS

	Chain C/Path 4	Chain E/Path 5	Chain F,G,M,N / Path 6,7,8,9
UNIT	TESTCEUC TESTCEUZ TESTCFUZ TESTMESU	TESTCEUE - DBX	TESTCEUF – DBX TESTCEUG – DBX TESTCEUM – DBX TESTCEUN - DBX
	DB = sssUDB04	DB = sssUDB05	DB=sssUDB06/07/08/09
	QUAL= aaaADU4	QUAL= aaaADU5	QUAL=aaaADU6/U7/U8/U9
	GRP = sssAPU4	GRP = sssAPU5	GRP=sssAPU6/U7/U8/U9
	PLAN= sssONLU4	PLAN= sssONLU5	PLAN=sssONLU6/U7/U8/U9
String	TESTCESC TESTCESZ TESTCFSZ TESTMESS	TESTCESE - DBX	TESTCESF - DBX TESTCESG – DBX TESTCESM – DBX TESTCESN - DBX
	DB = sssSDB04	DB = sssSDB05	DB=sssSDB06/07/08/09
	QUAL= aaaADS4	QUAL= aaaADS5	QUAL=aaaADS6/S7/S8/S9
	GRP = sssAPS4	GRP = sssAPS5	GRP=sssAPS6/S7/S8/S9
	PLAN= sssONLS4	PLAN= sssONLS5	PLAN=sssONLS6/S7/S8/S9
CAT	TESTCECC TESTCECZ TESTCFCZ TESTMESC	TESTCECD - DBX	TESTCECF -DBX TESTCECG –DBX TESTCECM –DBX TESTCECN –DBX
	DB = sssCDB04	DB = sssCDB05	DB=sssCDB06/07/08/09
	QUAL= aaaADC4	QUAL= aaaADC5	QUAL=aaaADC6/C7/C8/C9
	GRP = sssAPC4	GRP = sssAPC5	GRP=sssAPC6/C7/C8/C9
	PLAN= sssONLC4	PLAN= sssONLC5	PLAN=sssONLC6/C7/C8/C9
Train		TESTCETD – DBX	

TESTCE** are DBT unless noted

sss = System Acronym (e.g. MCP, PAE, NAS, etc.)

aaa = Department Acronym (e.g. DSS, NSP, DAS, etc.)

Chain F,G,M,N / Path 6,7,8,9 used by Medicaid only for DBX

**********Additional TESTCE** agency specific*******

Roads – TESTCETR,TESTCECR, TESTCESR, TESTCEUR

Labor - TESTCECL, TESTCESL, TESTCEUL, TESTCETC

Patrol - TESTCEP1

Emergency - TESTCEEA

APPENDIX C

AUTHORITY MATRIX

SYSTEM AUTHORIZATIONS

Note: this section was removed. See DB2 catalog for current authorizations.

APPENDIX D

CCF/MMF MIGRATION

NORMAL MIGRATION

When a request is made to move a program from CAT to STA (stage), the following actions must take place:

1. The SOURCE must be moved to STA.SOURCE from one of development libraries:
   a. CAT.CHAINx.SOURCE, where (currently), x is A through F

2. The LOAD MODULE must be moved from:
   a. CAT.CHAINx.BAT.LOADLIB to TPD.BAT.LOADLIB
      or
      CAT.CHAINx.OBJ.LOADLIB to TPD.OBJ.LOADLIB
      or
      CAT.CHAINx.CICS.ESA.LOADLIB to TPD.CICS.ESA.LOADLIB

3. The Data Base Request Module (DBRM) must be moved to TPD.xxx.DBRMLIB from the DBRM library (xxx = BAT/ONL/OBJ):
   a. CAT.CHAINx.xxx.DBRMLIB, where (currently), x is A,B,C,D,E,F,G,M,N

4, The SOURCE which was moved to STA.SOURCE must be compiled creating both of the following:
   a. A DBRM in STA.BAT.DBRMLIB / STA.ONL.DBRMLIB
   b. A Batch LOAD in STA.BAT.LOADLIB
      or
   c. An Online LOAD in STA.CICS.ESA.LOADLIB.

5, The PRD DBRM and LOAD must be copied from PRD to PRE (first).

6. The STA DBRM and LOAD must be copied from STA to PRD (second).

7. A Package Bind must be done in production using the STA DBRM.

8. A Package Bind must be done in every test STR(string) and CAT region, and in all training regions, using the concatenation of the CHAIN and TIER DBRM library, of that region and above, and the TPD DBRM library.

NORMAL MIGRATION OF SOURCE AND MODULES

DEVELOPMENT

CAT	DBRM	SOURCE	LOAD
TPD	DBRM		LOAD
	MOVED		MOVED

PRODUCTION

STA	DBRM	MOVED	LOAD
(1)	CREATED	COMPILED	CREATED
PRD	DBRM		LOAD
	COPIED		COPIED
(3)	from STA 2nd		from STA 2nd
PRD	DBRM		LOAD
	COPIED		COPIED
(2)	from PRD 1st		from PRD 1st

SUMMARY of NORMAL MIGRATION

1. The LOAD and DBRM in CAT are moved to TPD and become the basis for all executions and binds for all levels in the Test and Training environments.

    3. Only the SOURCE is moved to STAGE where the precompiled and the link create new LOAD and DBRM which are the basis for all executions and binds in the Production environment. The precompiled uses the 'VERSION (AUTO)' feature.

        ALL PRODUCTION MODULES ARE CREATED FROM ACTIONS WHICH START WITH THIS SOURCE.

3. The current Production (PRD) LOAD and DBRM are moved to the Previous (PRE) libraries.

4. The new Stage (STA) LOAD and DBRM are copied to the current Production (PRD) libraries.

EMERGENCY MIGRATION

An Emergency Migration is essentially a two phased migration strategy. The first phase is a migration to the QFX environment where a program can be executed in an emergency situation. The second phase of the migration involves the determination of whether or not the fixed program is to become the permanent version of the program, and the movement, creation, and deletion of modules to implement that decision.

PHASE ONE

1.   When a request is made to move a program from Emergency in the Test system (EMR) to Quick Fix in Production (QFX), the following actions must take place:
    a.   The Source must be moved from:
        ●   The Source in QFX.SOURCE must be compiled creating:

2.   A DBRM in QFX.xxx.DBRMLIB (xxx=BAT/ONL/OBJ)
    a.   A LOAD in:
    b.   QFX.BAT.LOADLIB
        or
        QFX.OBJ.LOADLIB
        or
        QFX.CICS.ESA.LOADLIB

3.   A Package Bind must be done in production using the QFX DBRM.

4.   The PRD DBRM and LOAD are NOT copied to PRE.

5.   The QFX DBRM and LOAD are NOT copied to PRD.

PHASE TWO

1.   If it is determined by the Manager that the current QFX modules are NOT to become the current Production modules, the following steps must be executed:

    a.   The QFX Source is deleted from QFX.SOURCE
    b.   The QFX modules are deleted from the QFX libraries in Production
    c.   The most current Package (QFX version) must be freed.
    d.   The EMR Source is deleted from EMR.SOURCE
    e.   The EMR modules are deleted from the EMR libraries in Test.

1. If it is determined by the Manager that the current QFX modules ARE to become the current Production modules, the following steps must be followed:

    a. The QFX Source is moved from QFX.SOURCE to STA.SOURCE
    b. The QFX modules are deleted from the QFX libraries in Production
    c. The most current Package (QFX version) must be freed.
    d. The EMR Source is deleted from EMR.SOURCE
    e. The normal migration process is initiated WITH THE SOURCE WHICH IS LOCATED IN THE STA.SOURCE (items #1 - #3 of a normal migration are bypassed).

    f. The Test Load module must be moved from:
       ● EMR.BAT.LOADLIB to TPD.BAT.LOADLIB
          or
          EMR.OBJ.LOADLIB to TPD.OBJ.LOADLIB
          or
          EMR.CICS.ESA.LOADLIB to TPD.CICS.ESA.LOADLIB

    g. The Data Base Request Module (DBRM) must be moved from:
       ● EMR.xxx.DBRMLIB to TPD.xxx.DBRMLIB (xxx=BAT/ONL/OBJ)
    h. A bind must be performed in all test STR(string) and CAT regions, and in both Training regions, using the concatenation of the CHAIN and TIER DBRM library and the TPD DBRM (CURRENTLY NOT BEING DONE).

EMERGENCY MIGRATION (Phase One)

TEST

EMR	DBRM	SOURCE	LOAD
	CREATED	COMPILED	CREATED
TPD			

PRODUCTION

QFX	DBRM	MOVED	LOAD
	CREATED	COMPILED	CREATED
STA	No Change	No Change	No Change
PRD	No Change		No Change
PRE	No Change		No Change

EMERGENCY BINDS (Phase One Showing DBRM Library Concatenation)

TEST

UNT	xxx.Chain?.xxx.DBRMLIB
STG	
CAT	
TRN	

PRODUCTION

PRD	PRODUCTION
QFX.xxx.DBRMLIB (xxx=BAT/ONL/OBJ)	

EMERGENCY MIGRATION (Phase Two)

TEST

EMR	DBRM	LOAD
TPD	DBRM	LOAD
MOVED	MOVED	

PRODUCTION

QFX		SOURCE	
STA	DBRM	MOVED	LOAD
(1)	CREATED	COMPILED	CREATED
PRD	COPIED		COPIED
(3)	from STA 2nd		from STA 2nd
PRE	COPIED		COPIED
(2)	from PRD 1st		from PRD 1st

SUMMARY of EMERGENCY MIGRATION

PHASE ONE

1.   Only the SOURCE is moved to QFX where the pre-compile and the link create new LOAD and DBRM in the appropriate QFX libraries. These modules are the basis for all executions and binds in the Production environment. The pre-compile uses the 'VERSION (QFX)' feature. (THIS OPTION IS NOT CURRENTLY USED)
2.   The current Production (PRD) LOAD and DBRM and the Previous (PRE) LOAD and DBRM are NOT moved or copied at this time.

PHASE TWO

If a Manager subsequently determines that the changes in QFX are to remain in production:

1.   The Source is moved from QFX.SOURCE to STA.SOURCE
2.   The modules in QFX are deleted.
3.   The LOAD and DBRM in EMR are moved to TPD and become the basis for all executions and binds for all levels in the Test environment.
4.   A normal migration is initiated starting with Step #4 of the Normal Migration Process.

BACKOUT and RECOVERY

A backout of changes, which have gone through the full migration cycle, can be accomplished by deleting the current PRD modules, and using the PRE modules according to the library concatenations.

THIS ACTION, HOWEVER, LEAVES ALL OTHER LIBRARIES IN AN INCONSISTENT STATE WITH THE SOURCE, BINDS, AND EXECUTION MODULES IN BOTH TEST AND PRODUCTION.

1. The following actions must be taken, after a full migration, to place all libraries back into a consistent state.
    a. The -1 SOURCE archive level in STA(stage) must be selected and saved as the current level. This action will force the backout version of the program source to be demoted to the -1 version.
    b. The LOAD MODULE in PRE must be copied to PRD.
    c. The LOAD MODULE in PRE must be copied to STA.
    d. The DBRM in PRE must be copied to PRD.
    e. The DBRM in PRE must be copied to STA.
    f. No Bind is necessary (the previous Package Bind will match the PRE(previous) (and now current (PRD)) Load Module).
    g. The SOURCE must be checked out from PRD into EMR.
    h. The SOURCE must be compiled in EMR.
    i. The LOAD MODULE and the DBRM must be moved to the proper TPD libraries.
    k. A Package Bind must be done in all test STR(string) and CAT regions, and in both training regions, using the concatenation of the CHAIN and TIER DBRM library and the TPD DBRM library.
    l. The SOURCE, LOAD, and DBRM must be deleted from the EMR libraries.

2. A program has been migrated to STA (stage), but has not been moved to Production, the following actions must be taken to place all libraries back into a consistent state.
    a. -1 SOURCE archive level in STA(stage) must be selected and saved as the current level. This action will force the backout version of the program source to be demoted to the -1 version.
    b. If the SOURCE has been compiled:
    c. The LOAD in PRD must be moved to STA(stage)
    d. The DBRM in PRD must be moved to STA(stage)
    e. No Bind is necessary (the previous Package Bind will still match the current PRD Load Module).
    f. If a Bind has been accomplished:
    g. The most current Package must be freed.
    h. If the program is going to be promoted again, no further action is required.
    i. If the program is NOT going to be promoted again, the following actions are required:
        *   The SOURCE must be checked out from PRD into EMR.
        *   The SOURCE must be compiled in EMR.

- The LOAD MODULE and the DBRM must be moved to the proper TPD libraries.
- A Package Bind must be done in all test STR(string) and CAT regions, and in both training regions, using the concatenation of the CHAIN and TIER DBRM library and the TPD DBRM library.
- The SOURCE, LOAD, and DBRM must be deleted from the EMR libraries.

BACKOUT

TEST

EMR	DBRM	SOURCE	LOAD
	CREATED	COMPILED	CREATED
TPD	DBRM		DBRM
	MOVED		MOVED

PRODUCTION

STA		SELECT −1	
	COPIED	SAVE	COPIED
(3)	from PRE 3rd		from PRE 3rd
PRD	COPIED		COPIED
(3)	from PRE 2nd		from PRE 2nd
PRE	No Change		No Change

TRAINING MIGRATION

1. When training is scheduled with the T/A group, a special AWP job is run. This job reinitializes the DB2 tables and rebinds all previously identified programs using the DBRM located in the TPD.xxx.DBRMLIB. (xxx=BAT/ONL/OBJ)

2. If Training Region 3 (TESTCETB) is to be used, and programs are to be used from the CAT environment, a request to bind the program(s) from either CAT Chain A or CAT Chain B must accompany the scheduling request.

APPENDIX E

DB2 MIGRATION/CONVERSION CHECKLIST

INTRODUCTION

The following document has been created to be used as a guide for the migration or conversion of programs or files which use DB2. While it is a guide, and can be modified as may be appropriate, the document is intended to be used to insure that all significant and important issues and concerns are addressed and considered in a typical DB2 migration or conversion.

This document is designed to be used by all agencies which require support from the Office of the CIO System DBA group. Based upon whether it is a (1) Simple Migration, (2) Complex Migration, or (3) Complex Conversion (the definitions are included in the Appendix A), separate sections have been prepared to highlight the information and actions needed to encourage a successful project implementation.

A second section, titled 'Test System Migration Worksheet' has been created for systems which use OCIO CCF/MMF migration process. Agencies which use their own methodology and testing scheme should develop their own test migration scheme. See Appendix B for a sample of this checklist.

In all cases, the document may be modified to meet special needs which may arise in a project. When modified or changed, those changes need to be communicated to all concerned to eliminate the chance of misunderstanding. When items are intentionally not included in a plan, explicit notation of the exclusion should be made.

As procedures are modified, or as additional subjects are found to be necessary or helpful, changes will be made to this basic document. In the event portions are found to be redundant or unnecessary, they can be nominated for removal.

SIMPLE MIGRATION or CONVERSION

This section should be used for the movement of new programs and/or files which are involved in 'READ ONLY' and/or reporting activity, but which do not update permanent data files.

PLANNING

Applications Development:	
SYSTEM PLAN (Collection) identified	
New program(s)/file(s)/control(s) identified	
CICS Support Notified (PPT, transaction)	
New file(s) defined	
GDG's defined	
DB2 Control Cards created and checked	
Operations documentation (APCDOC) complete	
JCL approved	
JOB schedule conflicts checked and resolved	
Dependencies and time of day	
Security authorizations identified	

Project DBA:	
SYSTEM PLAN created	
Plan Execution Authority granted	
New program(s) identified	
MVS Ids and authorizations identified	

IMPLEMENTATION

Applications Development:	
Manager has reviewed and confirmed the completion of all planning items	
Managers approval for Implementation	
Tests Reviewed and Approved	
Users Notified of Schedule	
Project DBA:	
Manager has reviewed and confirmed the completion of all planning items	
Managers approval for Implementation	
Execute Bind	
PLAN_TABLE Review(s)	
Plan or Package Execution Authorization	
DDF Table entries made	

POST-IMPLEMENTATION

Applications Development:	
Programs tested and functioning as expected	
Project DBA:	
DB2 PLAN active	

COMPLEX MIGRATION

This section should be used for the movement of new programs and/or files which are involved in the update of new permanent data files.

PLANNING

Applications Development:	
SYSTEM PLAN (Collection) identified	
New program(s)/file(s)/control(s) identified	
CICS Support Notified (PPT, transaction)	
New file(s) defined	
GDG's defined	
DB2 Control Cards created and checked	
Operations documentation (APCDOC) complete	
JCL approved	
JOB schedule conflicts checked and resolved	
Dependencies and time of day	
New DB2 Table Object(s)	
Tables Defined and Normalized	
New View(s) required	
Columns (defaults) Reviewed	
Relationships Verified	
Delete Rules Verified	
Identify Audit and Data Capture Requirements	
Table Authorization requirements defined	
DBM Notified of authorization requirements	
Security authorizations identified	

Project DBA:	
SYSTEM PLAN created	
Plan Execution Authority granted	
New program(s) identified	
New DB2 Table Object(s)	
Tables Defined and Normalized	
Columns (defaults) Reviewed	
Relationships Verified	
Delete Rules Verified	
Indexes Reviewed	
Table Authorizations Reviewed	
Audit and Data Capture Requirements Reviewed	
DDF Table entries made	
Space Calculations Completed	
STOGROUP and Volumes Reviewed	

DDL Generated	
Utilities JCL Generated	
DB2 Control Cards created and checked	
DB2 MIGRATION/CONVERSION CHECKLIST	
Image copies	
Platinum Stats	
Modify Recovery	
Log Analyzer	
APCDOC complete	
JCL approved	
New file(s) defined	
GDG's defined	
JOB schedule conflicts checked and resolved	
Dependencies and time of day	

IMPLEMENTATION

Applications Development:	
Contact List Prepared and sent to DBM and Operations	
Manager has reviewed and confirmed the completion	
of all planning items	
Managers approval for Implementation	
Tests Reviewed and Approved	
Users Notified of Schedule	
Populate new tables and Initial Load jobs	
Project DBA:	
Manager has reviewed and confirmed the completion	
of all planning items	
New DB2 Table Object(s)	
STOGROUP determined	
Data base created	
Tablespace(s) created	
Table(s) created	
Index(es) created	
Tables altered for Referential Integrity	
DDL Executed	
Data Base Authorizations for Utilities	
DDF Table entries	
Table Authorizations	
Utilities JCL Scheduled	
Application Initial Load(s) Complete	
Execute Bind	
PLAN_TABLE Review	
Image copies and other utilities	

POST-IMPLEMENTATION

Applications Development:	
Programs tested and functioning as expected	
Data verified	
Project DBA:	
DB2 PLAN active	
DB2 Data Base and Tables available (no flags)	
Tables Structures verified	
Applications Notified	

COMPLEX CONVERSION

This section should be used for the movement of changed and new programs and/or changed and new files which are involved in the update or the storage of new or changed permanent data.

PLANNING

Applications Development:	
SYSTEM PLAN (Collection) identified	
New program(s)/file(s)/control(s) identified	
CICS Support Notified (PPT, transaction)	
New file(s) defined	
GDG's defined	
DB2 Control Cards created and checked	
Operations documentation (APCDOC) complete	
Change to Package Bind	
Utilities Changes identified	
JCL approved	
JOB schedule conflicts checked and resolved	
Dependencies and time of day	
New DB2 Table Object(s)	
Tables Defined and Normalized	
New View(s) required	
Columns (defaults) Reviewed	
Relationships Verified	
Delete Rules Verified	
Table Authorization requirements defined	
DBM Notified of authorization requirements	
Security Changes identified	
Old DB2 Table Object(s)	
Table changes identified and documented	
View(s) changes required	
Columns (defaults) Reviewed	
Relationships Verified	
Delete Rules Verified	
DBM Notified of changes by object	
DBM Notified of authorization changes	
Backout Contingency Plan complete and reviewed	
Time selected for changes	
Business Users Contacted	

IMPLEMENTATION

Applications Development:	
Contact List Prepared and sent to DBM and Operations	
Manager has reviewed and confirmed the completion of all planning items	
Managers approval for Implementation	
Tests Reviewed and Approved	
Users Notified of Schedule	
New DB2 Table Object(s)	
Application Initial Load(s)	
Old DB2 Table Object(s)	
Application Conversion Program(s) Run	

Project DBA:	
Manager has reviewed and confirmed the completion of all planning items	
New DB2 Table Object(s)	
STOGROUP determined	
Data base created	
Tablespace(s) created	
Table(s) created	
Index(es) created	
Exception Table(s) created and/or altered	
Tables altered for Referential Integrity	
DDL Executed	
Data Base Authorizations for Utilities	
Table Authorizations	
Utilities JCL Scheduled	
Old DB2 Table Object(s)	
Unload Tables using DSNTIAUZ	
PLATINUM BP Executed	
Application Initial Load(s) Complete	
Execute Bind	
PLAN_TABLE Review	
Conversion Program(s) Complete	
New RI Required	
Image copies and other utilities	
DB2 Control cards modified	

POST-IMPLEMENTATION

Applications Development:	
Programs tested and functioning as expected	
Data verified	
Project DBA:	
DB2 PLAN active	
DB2 Data Base and Tables available (no flags)	
Tables Structures verified	
Data verified (Utility Row Counts)	
Applications Notified	
Verify Table Authorizations	
Verify Utilities JCL changed and moved	

CHANGE TYPES

1.   MIGRATION - the movement of programs or files, IMS data bases, or DB2 table objects, from test to production, which are new, and which do not effect any other program, file, IMS data base, or DB2 table object already in production.

2.   CONVERSION - - the movement of programs or files, IMS data bases, or DB2 table objects, from test to production, which are either new or changed, and which do effect other programs, files, IMS data bases or DB2 table objects which are already in production.

COMPLEXITY

1.   SIMPLE MIGRATION or CONVERSION   - the movement of new programs and/or files which are involved in 'READ ONLY' and/or reporting activity, but which do not update permanent data files.

2.   COMPLEX MIGRATION - the movement of new programs and/or files, which are involved in the update of new, permanent data files.

3.   COMPLEX CONVERSION   - the movement of changed and new programs and/or changed and new files which are involved in the update or the storage of new or changed permanent data.

MANAGERS

1.   MANAGER - a person who writes a performance evaluation or review on another person.

2.   SENIOR MANAGER - a person who writes a performance evaluation or review on another manager.