

PRISM Data Architecture Standards

*Standards for Database Administrators
and Developers
(**Appendices NOT Included**)*

Copyright 2007 DCPSA All rights reserved. No part of the contents of these materials may be reproduced, in whole or in part, in any form or media, without the prior written permission of the Legal Department of DCPSA.

Table of Contents *(Appendices NOT Included)*

About this Document.....	1
Introduction.....	1
Purpose.....	1
Database Standards	1
Personnel Responsible for Database Administration.....	1
Physical Standards	2
Database Releases.....	2
Testing Levels and Code Elevation	2
Stored Procedure, Function, Views and Triggers	3
Change Request Form.....	5
Database Maintenance	5
Server/Database Performance	5
Data Models.....	6
Model Standards	6
Database Objects.....	6
Tables	6
Fields	7
Indexes.....	8
Triggers	9
Data Types.....	9
User Defined Data Types	11
Defaults	11
Documentation.....	14
Schema Distribution.....	14
Database Environments and Production Access.....	14
Data Access.....	14
Transaction Processing	15
Auditing	15
User Issues.....	16
Security	16
Testing.....	17
Demos and Training.....	17
Technical Issues	18
Tools	18
Standard Operating Procedure	18

New User Additions.....	18
Middleware and Connectivity.....	19
Middleware	19
Connectivity	19
SQL Server Setup, Configuration and Operations.....	19
Setup and Configuration	19
Operations.....	20
Purchased Products	21
Appendix A.....	22
Server Configurations	22
System Hardware	22
System Software	22
Installation Configuration	22
Lines of Responsibility	23
Current Batch SchedulesLines of Responsibility	25
Database Creation and Refresh From Dump Procedures	26
Appendix B.....	27
Forms and Documents	27
Request Form.....	27
Stored Procedure Template.....	27
Setting Up Mail and Backup jobs on a new Server	27
Change Request form.....	28
Appendix C.....	32
Power Designer Procedures	32
Physical Model.....	32
SubModels	32
Table Editor	32
Column Editor.....	32
User Defined Datatypes	32
Referential Integrity	33
Development Database Changes.....	33
Test and Production Database Changes.....	34
Generating the Data Dictionary	35
Utilities.....	35
FKNameCheck.sql.....	35
AuditCheck.sql	35

(Appendices NOT Included)

List of Figures

Form 1: Request Form	28
Form 2: Stored Procedure Template	29
Form 3: Setting Up Mail and Backup jobs on a new Server.....	31

(Appendices NOT Included)

List of Modules and Action Verbs

About this Document

Introduction

The following PRISM database standard was adopted by DCPSA software development team. Following this standard will keep all our databases consistent. It is important that new PRISM database developments follow this standard, so our databases can easily be understood and integrated in the future.

Purpose

This document has been created with several purposes in mind. First, it is for the new DBA coming into DCPSA. It presents several of the policies and procedures we use in our every day operations, as well as recovery documentation.

Second, this document is to be used by people that directly interact with the data architecture team. There are development standards, security standards, and operating system standards built into this document.

This document will be in a constant state of flux, since we will, of course, update as our world changes. Any part of this work is open to discussion, and the data architecture team welcomes any input from other DCPSA employees.

Database Standards

Personnel Responsible for Database Administration

The DBA Team has the following responsibilities:

- Installing Microsoft SQL Server.
- Granting roles and permissions to SQL Server users.
- Managing and monitoring the use of disk space, memory, and connections.
- Creating, backing up and restoring databases.
- Diagnosing system problems.
- Configuring SQL Server to achieve the best performance.

- Perform day to day activities and management of the DBMS.
- Maintaining the physical data model in Power Designer
- Applying schema changes to various database environments
- Migrating static data from its source environment to various other environments

The DBA has the following authorizations:

- The only user(s) with access to the System Administrators (SA) role.
- The only user(s) who has the ability to change the SA password.
- Has sole direct access to production and test servers.

Physical Standards

Database Releases

- All changes to the physical design of a database (indexes, columns, tables, views, etc.) must be done through the DBA group.
- Changes to the physical structure of the database will be grouped, versioned, and tagged for promotion with corresponding application and static data releases. Version promotion will be executed in accordance with the Configuration Management Guidelines. (see change management plan)

Testing Levels and Code Elevation

- All database objects (schema, views, stored procedures, etc.) must pass through each testing level and be verified by the users of that level before being elevated to the next level.
- The progressive levels will be Development/Unit Testing, System/Integration Testing, Acceptance Testing, and Production.
- All promotions to production will be executed in accordance with the configuration management guidelines. (see change management plan)
- Only emergency code fixes will be applied to a production database during business hours. Any release code will be scheduled to be elevated during off-hours and verified at that time.
- Any exceptions to the last two rules must be passed through the testing group and approved by the project manager.

Stored Procedure, Function, Views and Triggers

- All script files should use the .sql extension
- SQL Code will be maintained in Source Safe outside of the database DDL.
- All Objects used in production will be owned by DBO.
- All SQL “select” statements should contain “NOLOCK” Clause where ever possible.
- All Stored Procedures should include “set nocount on” at the beginning and “set nocount off” and “grant execute” statement at the end.
- All stored procedures will contain begin and end transactions for processing.
- Errors that are called from stored procedures will be created and maintained in Alerts. Alerts that require interaction will send e-mail to a specific operator.
- Developers must keep a master copy (in Soucesafe) of any code they use. The database will not be used as a storage facility or a code source.
- Any code elevation must have a back out plan in place. If an elevation fails, the DBA will run the back out, inform the developer, and send the developer a list of compile errors.
- All temp tables created must be local temp tables (not global).
- All temp tables created in a stored procedure must be closed in that procedure, with extra error handling at the end of the procedure, double checking for the existence of the temp table and destroying it if it exists.
- All cursors must be closed and explicitly deallocated before the end of a procedure
- The following stored procedures are restricted in use:
 - xp_cmdshell – only used by sa
 - xp_sendmail – only used by sa
- Only single quotes should be used in stored procedures, as many tools follow the ANSI standard and cannot handle double quotes (or interpret them in a different way).
- All Objects will contain version tags in the header.

Naming Code Objects

Table Name is the name of the main related table and Optional Description is added for more clarity if needed.

Generation Code + Object Type Code + Module Code +
Group Code + Action Verb + Table Name + Optional Description

For example: zgtiAudit00InsertClientDrugTest
 cuspAgent00ScheduleClientDrugTest

Generation Code	Description
Zg	System Generated Code
Cu	Custom User Code

Object Type Code	Description
Sp	Stored Procedure
Vw	View
Fn	FN
Ti	Insert Trigger
Tu	Update Trigger
Td	Delete Trigger
Tiu	Insert and Update Trigger

Module Code	Description
Audit	Audit
Admin	System Administration
Agent	Agent
Client	Client
CA	Case Assignment
DI	Diagnostic Interview
RA	Risk Assessment
Rpt	Report

Group is defined by developer to allow grouping related code together:

Group Code	Description
00	Default Code
01	First Group
02	Second Code

Action Verb	Description
Insert	Insert records
Update	Update records
Delete	Delete records
Get	Get value

Change Request Form

- After an application or database has been placed into production, all application or database changes require that a change request form be submitted to the DBA Team.
- The change request form must be signed off on by the user requesting the change, the change manager or project manager responsible for the application and/or database, the programmer, and the DBA.
- The change request form consists of database names, the affected application names and a description of the changes to be made.
- The DBA Team maintains the change request file.
- Forms are located in Appendix B.

Database Maintenance

- All databases are completely backed up each night after a Database Consistency Check has been completed.
- The networking group will complete a backup of the entire system at the end of each night.
- Batch jobs will be run nightly to check database consistency and update database statistics for performance
- Accurate records will be maintained for server and database related changes and problems encountered by the DBA. This may be valuable information for subsequent DBA's managing the SQL Server environment.
- Backup and recovery procedures will be documented and maintained on-line and offline in the event of an emergency.
- All batch jobs will be scheduled through the DBA group.
- Any mirroring of data should be done through approved tools only, at approved times only.

Server/Database Performance

- Currently, the DBA support person is responsible for monitoring the production server during business hours.
- The tool used for database monitoring currently is Performance Monitor, which is built into Windows and SQL Server.

- SQL Server stored procedures will be debugged using Visual Interdev's SQL Debugger.

Data Models

Model Standards

- All modeling will be done using PowerDesigner 12

Database Objects

Naming conventions will be strictly followed:

Tables

- Table names need to be self-explanatory and should completely define table content without ambiguity.
 - All table names are singular.
 - Table names have uppercase letters between whole words. (e.g. ClientReleaseOrder)
 - The first letter of each word in a table is capitalized. The letters of acronyms are all capitalized, e.g. CJEntity
 - There will be no underscores in the table or field names.
 - All lookup tables will begin with a lowercase "lu".
 - All associative (link) tables will begin with a lowercase "lnk" and will indicate which tables are being linked together. (e.g. lnkClientDocketReleaseOrder).
 - Audit tables will begin with a lower case "a", followed by the exact name of the table they audit.
 - Client data containing tables will generally begin with the prefix "Client". There are some exceptions to this rule (e.g. CourtAppearance).
 - Definition tables will generally end in the suffix "Def". There are some exceptions to this rule (e.g. ReleaseCondition)
 - Security and Admin tables begin with the prefix "SA".
- Each table will have a primary key, which is a constraint. See Constraints for more detail.
- The primary key of lookup data tables will be a 10 character code field. This column will be named with the name of the table, less the "lu" prefix, plus the suffix 'CD'. (Example: Table luRace has primary key RaceCD). One important exception exists to this standard: The table luProgram has an identity key name ProgramCD. This is

the only "CD" column that is not a char 10. luProgram is used by the conversion program to audit the procedures performing inserts and updates in Create/AuditProgramID.

- Client data and definition data tables will have an identity column as an artificial primary key. This column will be named with the name of the table plus the suffix 'ID'. (Example: Table Client has primary key ClientID). It is not recommended that the primary key index be clustered, as this can result in "hot spots" and create performance problems.
- Rules and defaults will not be done as part of the table definition. Instead, they will be created independently and bound to the table.
- All tables that allow updates have a corresponding audit table. The audit table name is the base table name prefixed with "a". The audit table has the same structure as the base table, with an additional identity primary key column. The audit tables have referential constraints only to their parent base table. A copy of the "pre-update" base table record is inserted into the audit table every time an update is performed on the base table.

Fields

- Field names need to be self-explanatory and should completely define the column content without ambiguity.
 - Underscores are not used in column names.
 - If a column name contains more than one word the first letter of each word is capitalized, e.g. LastName, ArrestDateTime.
 - Identity keys are named with the table name + a suffix "ID". Two exceptions to this rule are: The UserID/OperatorID fields are varchar(20) and luProgram.ProgramCD is an identity.
 - 10 character code fields (lookup codes) are named with a suffix "CD".
 - 1 character indicator fields are named with a suffix "Ind" (e.g. BirthDeathInd is char(1), domain: "B,D"). These fields are used when more than 2 states are required or character data will be more meaningful than a flag; and not enough values are present to justify a lookup table.
 - Boolean values use the tinyint datatype and are named with a suffix "Flag". 1 = True, 0 = False. Defaults are defined as appropriate for each field. (e.g., CurrentFlag defaults to 1, LogicalDeleteFlag defaults to 0).
- Identity fields are used when there is a need to track the next available id or have the system create a unique number (including artificial keys).
- The data type of identity columns is *int*.
- The *text* data type is very inefficient and is discouraged when possible.

- Default and check constraints are placed on columns whenever appropriate.
- Character fields are defined as CHAR types when size of field is known or size of field is smaller than 10 characters.
- Character fields with sizes larger than 10 or unknown in size will use the VARCHAR data type.
- Data fields are designed to use the minimum amount of space acceptable.
- Fields left empty by user input are assigned default values when feasible.
- Fields being used for holding temporary data begin with a lower case “t”.
- The following audit & control columns are present on every table:

CurrentFlag	tinyint	Indicates current/prior data
AuditOperatorID	varchar(20)	Last modify user
AuditOperationDate	datetime	Last modify date
AuditProgramID	int	Last modify program
LogicalDeleteFlag	tinyint	Indicates deleted data
CreateOperatorID	varchar(20)	Create user
CreateOperationDate	datetime	Create date
CreateProgramID	int	Create program

CurrentFlag is defaulted to 1. LogicalDeleteFlag is defaulted to 0. All other audit fields are populated by the application code.

- Data is NEVER physically deleted. Updating LogicalDeleteFlag = 1 is performed instead of deleting.

Indexes

- Adequate performance testing has not yet been conducted to facilitate indexing strategies. Currently, only system generated indexes exist. Indexing strategy will need to be addressed as development and testing proceeds.
- Number is incremented sequentially beginning with 1.
- Frequently used key is index with default name of IX_TableName_ColumnName
- Primary Key Indexes are Unique and may be Clustered or Non-Clustered. If the primary key column is an identity column, the PK index should not be clustered to avoid “hot spots” (a performance difficulty that arises when inserts are contending for the same data page to insert new rows due to clustering on an identity value).
- Alternate Key Indexes are Unique and may be Clustered or Non-Clustered
- Foreign Key Indexes are Non-Unique and may be Clustered or Non-Clustered
- Inversion Entry Indexes are Non-Unique and may be Clustered or Non-Clustered

Triggers

- Trigger names begin with the letter “t” followed by “i” (insert), “u” (update), “d” (delete) or any combination followed by the table name, e.g. Insert/Update trigger on Client table would look like cutiuClient.
- All triggers will have a description of what their purpose is and any other tables called by them.
- All triggers will contain version tags in the header.
- All triggers will be maintained in SourceSafe, not as part of the data model.

Data Types

Int

- Any whole number field that can exceed 32767. The max value for an Int is 2,147,483,647.
- Identity Columns -- Int is the datatype for all Identity columns. It is used for all “ID” fields that have numeric values.

Smallint

- Any whole number field that will exceed 255, but will not exceed 32767.

Tinyint

- Whole numbers from 0 to 255. Tinyint is mainly for “Flag” data. Flags are usually binary questions with a null option for the absence of any answer. Defaults are applied to flags when null values are not allowed. The bit datatype was not used for Flag data to allow the null option when appropriate. Also, the staff has experienced some past problems with the bit datatype.

Bit

- Bit is currently not used.

Char

- Char is used when the field length is less than 10 or the exact length of the field is known and it is a non-nullable field.
- Char is also used when a field is part of a primary key so the PK value will not have to be compressed and decompressed when accessed.

Varchar

- Varchar is used for string data when the field length is greater than 10 or the exact length is unknown.

Datetime

- This is used for most date and time fields.

- Datetime can store date and time data from January 1, 1753, to December 31, 9999, with an accuracy of three-hundredths of a second, or 3.33 milliseconds.

Smalldatetime

- SmallDateTime can store date and time data from January 1, 1900, through June 6, 2079, with an accuracy of one minute.

Identity Columns

- All tables, except link tables and lookup tables, will have an IDENTITY column to uniquely identify each row.
- The datatype is Int.

Money

- All money fields will be stored as money for values that can be higher than 214,748 or lower than - 214,748.

Smallmoney

- Can be used for fields that will not require amount higher than 214,748 or lower than - 214,748.

Not Null

- Identity columns and fields which are part of a primary key must be defined as not null. Other data elements which are required according to business rules are defined as not null.
- Due to the use of artificial identity keys, foreign key attributes are not migrated as composites in the primary key of the child table, and therefore the declarative referential integrity is defined as optional. Non-optional foreign key references are defined as not null in the child table since this is not declared in the referential integrity.

Null

- Null is used for columns that do not require an entry based on business rules.
- Some exceptions occur, where data that should be required according to the business rules is defined as null. These exceptions are:
 1. Where order of processing within the application necessitates a required field to be null for a period of time during the process
 2. Where values are required in PRISM but may not be available for converted records. Many of these instances are required by the PRISM application.
 3. Where the business rule was not known at the time the database definition was performed. Many of these instances are required by the PRISM application.

User Defined Data Types

User defined data types will be self-explanatory and should completely define the data type content without ambiguity. A data type used to define a field in one table must then be used to define the same field in all tables.

Data types currently in use are:

User Defined Name – Microsoft SQL Server data type

- AuditOrgID - int
- AuditOperatorID - varchar(20)
- AuditOperationDate - datetime
- AuditProgramID - int
- CreateOrgID - int
- CreateOperatorID - varchar(20)
- CreateOperationDate - datetime
- CreateProgramID - int
- CurrentFlag - tinyint (default = 1)
- LogicalDeleteFlag - tinyint (default = 0)
- LegacyInputSequence - int
- PreTrialFlag - tinyint (default = 0)
- ProbationFlag - tinyint (default = 0)
- ParoleFlag - tinyint (default = 0)
- Remarks - varchar(255)

Defaults

Rules and defaults will always be bound to a table by using the sp_bindefault stored procedure, never as part of the table definition. Defaults will be self-explanatory and keep current in this document.

Current Defaults are:

- Set To Zero - The value is 0.
- Set To One - The value is 1.

Naming and Ordering

- 1) Table or Column once created will not be renamed or dropped
- 2) New Column of a table will always added at the end of the table.
- 3) TableName will begin with Module Code + a Singular Descriptive Name

Module Code	Description
A	Audit table of the corresponding table
CA	Case Assignment
Client	Client
DI	Diagnostic Interview
Lu	Lookup Table
Sa	System Administration

- 4) Table created in PRISM will begin with these columns:

 TableName

TableNameID	int	Identity, Primary Key
CurrentFlag	tinyint	Indicates current/prior data
LogicalDeleteFlag	tinyint	Indicates deleted data
AuditOperatorID	varchar(20)	Last modify user
AuditOperationDate	datetime	Last modify date
AuditProgramID	int	Last modify program
CreateOperatorID	varchar(20)	Create user
CreateOperationDate	datetime	Create date
CreateProgramID	int	Create program

 Other Columns

- 5) Each table created in PRISM will have a corresponding audit table with a prefix “a” For example table Client, its audit table will be aClient. Audit table will have all the columns of the original table plus it’s own Primary Key

 aTableName

aTableNameID	int	Identity, Primary Key
TableNameID	int	
CurrentFlag	tinyint	Indicates current/prior data
LogicalDeleteFlag	tinyint	Indicates deleted data
AuditOperatorID	varchar(20)	Last modify user
AuditOperationDate	datetime	Last modify date
AuditProgramID	int	Last modify program
CreateOperatorID	varchar(20)	Create user
CreateOperationDate	datetime	Create date
CreateProgramID	int	Create program

 Other Columns

- 6) Each table must have a primary key named PK_TableName, all foreign keys will be named as FK_TableName_ColumnName for example:

 Table ClientName

PK_ClientName is Primary key with key ClientNameID

FK_ClientName_ClientID is foreign Key with key ClientID

- 7) Index is named as IX_TableName_ColumnName1_ColumnName2 for example:
an Index of ClientName for composite keys of FirstName, LastName and MiddleName is
X_ClientName_FirstName_LastName_MiddleName

Documentation

Schema Distribution

- All models will be stored in source safe.
- The Data Modeler/DBA will make all changes.
- The Data Modeler/DBA will make data dictionaries available on a shared network drive.
- The Data Modeler/DBA will provide printed ERDs upon request.

Database Environments and Production Access

- No developers should have access to the production box.
- A replicated architecture is planned for the production environment.
- If reporting is required to be done in a separate database or server due to performance issues, a reporting environment should be established and a scheduled job should regularly refresh the environment from production. The tool for this process has not been determined, however DTS may be an appropriate choice.
- Acceptance test, system/integration test, and development environments should be constructed according to configuration management guidelines. These environments should not be copied from one another. They may be created with a copy of the production environment in order to test the promotion of a codeline.
- The following environments are currently in use (database server / application server):
 1. PRISM Production: APPVSQL.PRISM / PSAPRISM1/2/3
 2. Development: PSADDEV1.prism / PSADDEV1
 3. PPUG User Testing: PSADTEST.PRISM / PSADTEST
 4. User Training: PSAP2TRAIN.PRISM/ PRISMP2TRAIN
 5. Developer Testing: PSADDEV1.PRISM / PSADDEV1

Data Access

- There will be no direct data access for any users of the databases. All data access must be done through stored procedures or the Microsoft Transaction Server.
- Developers will code any stored procedures that involve business rule processing.

- Large reports, interfaces, or other batch processes should be run outside of business hours whenever possible.

Transaction Processing

- Enterprise locking strategies have not been defined. Update vs. read volumes should be analyzed if current locking approaches are found to be inadequate in load testing or production. For details on locking strategy being employed by MTS, see PRISM Development Standards.
- All stored procedures will contain begin and end transactions for processing.
- Business rules and transactions will be handled via Microsoft Transaction Server.

Auditing

- Audit fields are a part of all tables. Referential Integrity is not enforced on audit fields.
- CreateOperatorID - UserID of the operator that entered the record. This value is never updated.
- CreateOperationDate - Date record was initially created. This value is never updated.
- CreateProgramID - ProgramCD (from luProgram) of process creating the record. This value is never updated. (Currently only populated by conversion procedures)
- AuditOperatorID - UserID of the operator modifying the record
- AuditOperationDate - Date record was modified
- AuditProgramID - ProgramCD (from luProgram) of process modifying the record. (Currently only populated by conversion procedures)

User Issues

Security

- A review of standard vs. integrated security will be done prior to production implementation. There may be a combination of integrated security and standard security used based on the types of applications being used. For development purposes we will begin using integrated security and SQL Server Roles.
- When using integrated security, no one with local administrative rights on a SQL server may change the password of SA unless that person is a member of the DBA group and has gone through the password changing procedure.
- Database Owner (SA/DBO/DBA) authority will only be given to those in the DBA group.
- User groups and roles will be universally used. As new databases are developed, universal groups will be created.
- Under integrated security passwords will be controlled via the users Window account. Under standard security passwords will be given a common initialization, and it will be the applications responsibility to force a password change upon first logon.
- The “Guest” user will be removed from all databases except master. Guest can not be removed from the master database based on Microsoft’s standards.

Application User

- PRISM Application will use account UserPRISM for Database connection

User Roles

- Roles will be reviewed and assigned at a later date.

User Access Form

- A user access request is to be submitted to the DBA Team for all users of applications and/or databases. The change request form can be filled out for new user additions.
- The user access request must be signed-off by the intended user, the program manager, and the DBA.
- The user access request consists of the security profile of the user, the type of access requested for each database and table, and the group the user is to be placed in.
- Any changes or additions to a user profile requires an update to the user’s access form.

- Program managers are responsible for notifying the DBA of any personnel changes and when personnel leave so that the users can be removed from the system.
- The database team attempts to keep a current list of which databases have interdependencies, but the user request form should indicate all database accesses needed.

Testing

- All testing must follow the standards given by the DCPSA.
- Development testing will be done only on the development server.
- Currently, stress testing is performed by the development group. See the development manager for details on the stress testing tool being used.

Demos and Training

- Demos and training should be done on a stable box – currently PSAP2TRAIN is used as the PPUG user test system.
- All demos and training should be cleared through the DBA group to reduce scheduling conflicts or possible problems.

Technical Issues

Tools

- The data modeling tool in use is Power Designer 12.
- Visual Studio.
- SQL 2005

Standard Operating Procedure

- Any procedure promotions must be requested far enough in advance that they can be properly screened for impact by a DBA.
- Weekend, night, and early morning hours will not be scheduled but may be pre-arranged.
- All requests must be submitted by e-mail, through the proper chain of authority, using the form provided.

New User Additions

- DCPSA security should be notified of any new user additions before the DBA team creates a SQL login of any type.
- Requests for new user additions will be taken only from those with proper authority.
- Only valid users will be added to production systems (no dummy accounts).
- User logon is generally the same as their window logon or 7 letters of the user's last name and first initial.
 - CLINTONW
 - BUSHG
 - JEFFERST
- The master database should be set as a user's default, while permissions to master should be removed.

Middleware and Connectivity

Middleware

- See PRISM Development Standards for details regarding MTS and Visual Basic.
- Users can currently connect via Named Pipes or TCPIP.

Connectivity

- Access *may not* connect to the databases.
- Power Designer can be used to apply changes to the development database. Changes beyond development should be made by performing a compare against the promotion region and generating an alter script to be checked into SourceSafe and subsequently executed in Enterprise Manager against the target database. All new databases will be created using scripts taken from SourceSafe (generated via forwarding engineering from Power Designer).

SQL Server Setup, Configuration and Operations

Setup and Configuration

Servers

- The Character Set is using the default ISO 8859-1.
- Sort Order is using Dictionary Order, Case Insensitive.
- Network support of Name Pipes and TCP/IP.
- Licensing Mode will be set “PER SEAT”.

Databases

- The master database is to be set to a default size of 100MB. The recommended Master Database set up is 20% larger than the required size to allow for growth.
- Each database will be optimized for the system that it is running on, filegroups will be created to take advantage of the system’s hardware and memory configurations. Currently, all tables are contained in the PRIMARY filegroup.
- When possible, the data device and the log device should be mirrored.
- Database files/filegroups are created after a determination is made on the amount of space necessary to hold records, indexes (clustered and non-clustered), images/text and views.
- Files/filegroups will be set to automatically increase space based on a set number (100mb) not with the percentage option.

- Auto shrink is turned off as the default for the Server version and will stay that way.
- When shrinking the clustered index, reorganization will also be run for that index.
- All development of databases, tables and applications by programmers is done on the development server.
- All conversion development will be done on the conversion server.
- Before an application or database is moved into production, it is tested on the system/integration test and acceptance test servers.
- The System Administrator, “SA”, is the owner of all databases unless otherwise directed by management.
- Owners of databases are required to follow the policies written and provide the Database Administrator notification when database changes are made. This notification will also provide copies of the scripts to create those changes. Failure to do so may prevent recovery of the database after a system failure.

Transaction Logs

- Logs will be set at a reasonable size to avoid constant autogrow.
- Autogrow will be set to grow in MB increments, do not use the percentage option.
- Log shrinking will be done manually.

TempDB

- Autogrow is turned on for TempDB.
- Autogrow will be set to grow in MB increments, do not use the percentage option.
- TempDB will be set on a fast I/O subsystem to get good performance and multiple files. Raid 1 is the most preferred.

File/Filegroups

- Each SQL server database has at least one data file, at least one log file and at least one dump device.
- All levels of a database must have files/filegroups in synchrony.
- Files/Filegroups will be reviewed and changed as the system/program needs change during the initial phase. Currently all tables reside in the PRIMARY filegroup.

Operations

Backups

- All databases are completely backed up each night after a Database Consistency check has been completed.

- The network group makes a complete backup of the entire system at the end of each night.
- A backup to disk three times daily is recommended for production databases.

Logs

- Transaction logs are truncated at time of checkpoint on all databases. This allows for no recovery from logs. This will be reviewed based on the size of the database and the need for recovery.

Scripts

- Full build and alter scripts are maintained in SourceSafe as components of release packages. Creation of any database/software release will be possible with release packages from SourceSafe.

Nightly Operations

- There will be a series of jobs that are run each night to update data and check consistency of databases (see Appendix A).
- SQL Mail will be setup to send e-mail about completed and uncompleted nightly operations.

Notifications Required for Use of SQL Server

- All system notifications are to be sent to the DBA Team.

Notifications to Users

- Currently, notifications to users are sent via e-mail.

SQL Administration and Maintenance

- Changes beyond the development environment will be made using SQL scripts in accordance with the configuration management guidelines (see Change Management Plan). All database changes and scripts will be the responsibility of the DBA team.

Purchased Products

Power Designer to support SQL Server 2000/2005.

AdeptSQL Diff