THE PROJECT CONTROL DATA WAREHOUSE MODEL (PCDW) v1.02

Cliff Brandon
June 12, 2014

AN INTRODUCTION TO PROJECT DATA WAREHOUSES

Updated By: Cliff Brandon 6/12/2014

TABLE OF CONTENTS

PROJECT CONTROL DATA WAREHOUSE NOTICE

License:

The contents of this file in either electronic or hard copy formats is subject to the Project Control Data Warehouse License Version 3.0 (the License). You may not copy or use this file, in either source code or executable form, except in compliance with the License. Review of this document or related source code, in whole or in part, constitutes acceptance of the terms and conditions of the License. You may obtain a copy of the License at http://www.asaservices.biz

Software distributed under the License is distributed on an AS IS basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

Copyrights:

Acknowledgements:

Special thanks to everyone who has made suggestions on how to improve the early versions of the product.

Modifications & Release Notes:

Versions: Base v.00 (Beta) 2009-03-31

v.01 (Beta) 2009-07-10

v1.00 2014-05-28

v1.01 2014-06-01

v1.02 2014-06-12

See the 'delta' report for release notes.

EXECUTIVE SUMMARY

The Project Control Data Warehouse (PCDW) is an instance of the Open Data Warehouse Model (ODWM). The central premise of the ODWM is that, for a given practice area, there is a model that best addresses the majority of information needs within the practice area. The PCDW is designed to address the Project Control practice.

INTRODUCTION

This document describes the boundaries and general arrangement of the Project Control Data Warehouse (PCDW). It is intended to be a fairly succinct technical document. To explore some of the rational used in constructing the PCDW, please refer to the *Introduction to Project Data Warehouses* white paper. Appendix A contains a summary of the PCDW features.

This is a living document and may be adjusted from time to time. Please note the version to determine if you have the most recent document. Generally speaking, the document version will match the product version. The License version is generally incremented in whole numbers and will not match the product version. Also note that this paper does not contain the actual PCDW, its data dictionary or license except by reference (see the Other Documents section).

AUDIENCE

This document is a technical document directed primarily at developers and solution providers who want to build extensions to the data warehouse (ETL Bridges, Reporting, OLAP cubes, etc.). Owners or information consumers may find it useful for understanding what kind of information can be available from this resource.

A moderate understanding of data modeling, BI/Data Warehouse, and SDLC is assumed. Knowledge of the Project Control practice area is helpful though not required.

OTHER DOCUMENTS

There are several other documents you might find helpful. All are available at http://www.asaservices.biz

License: The PCDW license.
PCDW: The actual PCDW model (script)
PCDW Data Dictionary: Description of all the elements of the Project Control Data Warehouse.

PCDW Delta report: Change management – differences between versions.
Introduction to Project Data Warehouses: Paper describing data warehouses for Project Control.
Commercial Subscribers will have a more extensive set of documents available.

Other documents will be made available as the need arises. These may be generally available to the community or may be available on a subscription basis.

ROAD MAP
After the initial release of the PCDW in March of 2009 the first 6 months were spent growing the community, creating ancillary documentation, developing additional tools, recruiting vendors, and tuning the system. During this time the PCDW was available as a beta release.

After the first open source style 'production' release (PCDW v1.00) in May 2014, a commercial version will be available. The 'open source style' data warehouse will continue to be developed after the commercial release.

The first commercial version of the PCDW (PCDW vC-1.02) is now available.

COMMERCIAL VERSION (PCDW-C)
The commercial version includes the model, dimensions, relationships, dimensions, extended descriptions (attribute definitions), and multiple work spaces. In addition to the actual model, additional documentation will be provided. Users of the commercial version will have access to a more specific product road map.

OWNER's and CONTRACTOR's POINTS OF VIEW
Owners and Contractors have different points of view/needs for project information. The PCDW attempts to satisfy the needs of both groups.

THE PROJECT CONTROL PERSPECTIVE
Project control is about answering one question, "How is the project going?", from many points of view. The nature of this question and the needs of the information consumers is what drives the PCDW design.

THE PCDW PRACTICE AREA

All organizations may be broken down into 3 functional elements:
Operations: Revenue Generator
Projects: Change Mechanism
Administration: Directs the organization including Operations and Projects

The PCDW focuses on the Project element for all types of organizations. Specifically, the PCDW is a repository for project information. The PCDW goal is to answer project progress questions from many points of view.

All organizations have projects. Not all organizations need the PCDW technology and none of them need it for all projects. Organizations with 'high risk' projects and revenues that can support the maintenance and evolution of a PCDW are good candidates for this technology.

DISTRIBUTION

Beginning with the release of PCDW v1.00, organizations which might want to embed all or a portion of the PCDW model in their product or reference the model will need a commercial license to do so. Otherwise there are few limits on what organizations or individuals may download and use the PCDW. The limitations and conditions of use are provided in the license.

PCDW BOUNDARIES

The PCDW is a data warehouse framework. It does not include any content except that which is included in the license table and some limited example data that may, at some point, be included with the model.

The PCDW v1.00 model does not include any ETL scripts, staging table or other tools for importing data. The model does not include any views, stored procedures or indices (other than those for primary keys) as these may not exist or may need to be optimized for various platforms. Likewise, security mechanisms or schemas are not included. These data warehouse elements are the responsibility of the implementers and/or data warehouse content owners.

The model does not include OLAP cubes, Repositories, BI tools, Reports, or analytic tools. However, warehouse 'bundles' for specific

sources, BI, Reporting, and Analytic solutions may be made available (at a cost) either by ASA or other Vendors.

The model does included script(s) to instantiate the warehouse. Documentation for the warehouse is included (general data dictionary and dimension documentation) along with a document identifying changes from the previous model.

UNDERLYING ASSUMPTIONS

The underlying assumption for the PCDW is that all projects have the same basic information needs, that those common needs are discoverable and the data to satisfy those needs can be collected in a common repository.

Having said that, there are conditions that might limit or preclude use:

1) Organizations may not have all the data or have 'clean data' necessary to develop a complete project picture, and
2) The data may be available but protected by proprietary systems, business processes, or non-standard systems.

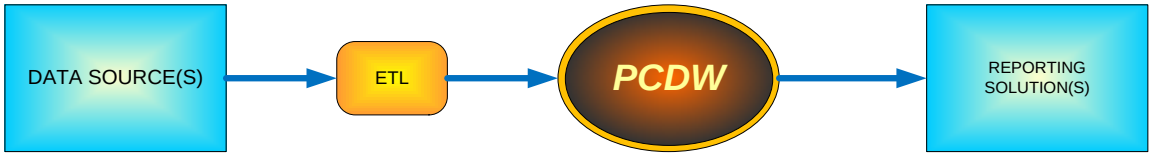
It should be noted that the initial beta models are approximations that will have to be refined over time with input from the community.

All organizations with sufficient project needs and access to their source data can use the model. There may be some special needs however that can't be accommodated by the model that will require the model to be adjusted or other mechanisms will need to be created to handle those needs (e.g. OLAP cubes, other aggregates, etc.). In these cases, you may want to consider creating these elements from data in the PCDW model or request model changes (chances are if you are having a problem, others are having similar problems).

PLATFORMS

The PCDW is designed to be database agnostic. It is designed to work on any modern SQL 92 compliant data base including 'open source' type data bases. It is designed to receive data from multiple sources. Any BI or other reporting or analytic tool that can attach to the database where the PCDW is loaded can be used. It is designed to be a scalable anything-to-anything solution.

Anything to Anything Soultion...



PCDW SCHEMA and PROPERTIES
The PCDW schema can best be described as a snowflake. There is a central table containing transactions. Dimension tables are linked to the fact table transactions via surrogate keys. Candidate keys (contains original natural keys) are included with most of the dimension tables.

A FEW SPECIAL TABLES
A license table is included to record copyright information for the PCDW. This is the only table provided that is 'pre-populated'. The License table is not linked to any of the PCDW tables.

A 'source' table is included to permit the tracking of original source information.

A number of authority tables (lookup tables) have been provided to permit terminology resolution between sites. An optional table has been provided that allows all authority tables to be consolidated into a single table.

THE 'KITCHEN SINK' ARCHITECTURE
Many data warehouses begin life as content heavy affairs that seem to have everything including the kitchen sink included in their architecture. This is not generally considered a 'best practice' and, if used initially, is generally trimmed, rearranged, and refined in the first or second round of development. After the initial rounds of refinement however, the architecture commonly begins to expand again to accommodate new needs.

At first glance the PCDW appears to have succumbed to this bloat. It is indeed big. There are some reasons for this and some reasons to resist the urge to immediately trim it to fit a specific need:

- The PCDW is a general model meant to accommodate a number of different needs and information sources. It is 'built big' to accommodate different project types for different organizations

(EPCM, IT, etc.) and allow for growth as organizations change and expand their interests. The PCDW It represents an expanded and tuned environment built from experience.

- The PCDW is meant to be a stable ETL target primarily for development. In some cases, you may be able to add to the architecture but trimming it may break the ETL process and defeat commercial ETL products developed for this environment.
- The project environment is naturally dimension rich. The PCDW tends to have a footprint similar to financial data warehouses in that it is 'wide' not 'deep'. Because the data sets tend to be smaller, there is a smaller penalty (load, search, etc.) for wide tables. If efficiency is a consideration, OLAP cubes and Data Marts can always be constructed from the PCDW content.

Normally, this would be the place to discuss tradeoffs but there aren't any. It's a win-win for you the information owner.

Option 1: You can take the model and tune it for your own specific needs (the custom approach) recognizing you are going to have to develop and tune everything using your own resources (ETL, BI, and other tools).

Option 2: Take the model and buy ETL components, tools, BI/Analytic solutions (there are open source options here as well). This 'standards' approach is a lower cost more flexible approach.

GRAIN

Each order, receipt, invoice, payroll, JV, estimate, schedule and other transaction is represented in the fact table. Depending on the type of transaction (budget, commitment, incurred, payment, etc.), blocks of fields are available for the transaction values.

HIERARCHIES

There are ragged, fixed, and date hierarchies. There is one notable 'ragged' hierarchy for projects that includes a mechanism for negotiating hierarchies recursively or via a bridge table. Each of the hierarchies will be discussed in the dimension documentation.

SLOWLY CHANGING DIMENSIONS

It is possible to make effective use of slowly changing dimensions. Where slowly changing dimensions can be used effectively, they are discussed in the dimension documentation.

It is important to remember that the principal goal of the PCDW is to determine project status 'now'. It is important to keep that in mind when considering implementing a slowly changing dimension. For example, you could keep a history of order changes. However, the 'now' context indicates that only the most current order will be considered. Where this might be an issue, it is discussed in the dimension documentation.

RECONCILING 'CODES' BETWEEN SOURCES

If or when the PCDW is used in a confederation of PCDWs for consolidated reporting, it is very likely that different groups will use different 'codes' for their 'lists' (eg. Action Codes, status, cost types, project types, etc.). Ideally, the various sets of 'codes' should be rectified to an organizational standard. In advance of a global reconciliation of 'codes' the PCDW provides a mechanism for retaining original lists and rectify them to a reporting standard.

This is accomplished using tables that map source list elements to an organization standard. For example, each source may have a list of employee types (e.g. Salaried, Hourly, Exempt, Non-Exempt, etc.). The titles may be different but select items have a common meaning. To resolve the 'meanings' to a single set of standards, a table is provided with records for each of the items on each of the lists (including the standard). A standard key allows each item on the source lists to be mapped to a standard. When attached to a dimension that utilizes the list, the codes from the various sources can be referenced in the dimension while the standard in the list can be used to group all similar references.

This technique is employed extensively in the architecture (primarily in authority files – lookups) and, with a few exceptions, each of these tables have similar structures.

It is possible to compile all the lists into a single list with the addition of a single 'list title' field. A table is provided (DAF_LISTS) for this should you choose this approach. You may still need to create views and aliases to accommodate this approach. Check your BI/Reporting tool to determine the best approach for your system.

THE 'FLEXIBLE STAR'

To accommodate optimization for the many possible combinations of Database and BI/Analysis tools the PCDW provides a 'flexible' star model. This model allows select 'flake' elements to be attached either

to another dimension or directly to the fact table. Database selection and tools (BI/Analytics) will likely drive your preferences.

Consider a typical order where the actual order transaction information is stored in the fact table and order header and detail information are dimensions. The dimensions can be attached in a star or snow flake arrangement [Order Header>Order Detail>Fact], or as a star [Order Header>Fact and Order Detail>Fact]. As an option, the Order Header data and aggregations from the detail, can be included in the Order Detail (denormalized) and then attached to the fact table. Each approach has advantages depending on the table size, database, and BI/Analytic tools used.

Relationships depicted on the model represent only one way in which domains could be associated with the fact table. The 'flexible star' allows other combinations. Note that the relationships between entities are not included as part of the model.

THE PCDW DATA MODEL

The data model is maintained using a data modeling tool. Code for creating the tables is also generated for the various data platforms using the modeling tool.

Beta version: 90 dimensions, one fact table, and ~3000 attributes.
PCDW 1.00: 86 dimensions, one fact table and ~3418 attributes.

The table and field descriptions are necessarily brief and don't make assumptions about business process.

OTHER PCDW FEATURES

The PCDW is primarily a set of dimensions with one fact table. It does not contain OLAP cubes, metadata repositories, ETL scripts, staging tables, indices, aliases, views, BI, reporting, or analysis solutions. All of these elements may vary by platform or may be implemented by the owner over time or as need dictates.

NAMING CONVENTIONS

The naming conventions used in the PCDW have 2 goals: 1) to identify, as clearly as possible, what the content is, 2) to link a field back to its original table.

NAMES, IDENTIFIERS, and KEYS

We have a lot of ways of naming physical and abstract objects (formal names, codes, acronyms, abbreviations, etc.). We aren't always consistent and uniqueness is a problem. How many 'instances' of John Smith do you think exist? Human beings can live with or adjust to naming inconsistencies but databases generally can't.

In a model as big as the PCDW (3000+ attributes) developers have to understand the naming 'patterns' used in the model.

Keys: Primary and foreign keys are surrogate keys (usually numbers) are how the relational system recognizes uniqueness and membership.
Identifiers: These are the 'codes' we (humans) use to identify 'objects'. For example, cost engineers will more likely remember a cost 'code' than the actual name or title of a cost code.
Names: We may know an individual as 'John Smith' but we are not likely to call him by his employee identifier or 'key'.

The following are conventions used to identify when an attribute is being used as a key, identifier, or name:

Type	Suffix - Pattern	Cost Code Example
KEY	_PK, _FK	afccCostCode_FK
IDENTIFIER	xxxId_NK, xxxId	afccCostCodeId (see note)
NAME	None	afccCostCode

Note: There are some exceptions where there either are no 'names' (Date) or the code is equal to the name (unit of measure).

TABLE NAMING

All fact and dimension tables follow the same naming convention (example):

DXX_AAAAAAA (e.g. DAF_COSTCODE: Cost Code Authority File)

All characters are in upper case to help distinguish them from field names. The leading character is always a 'D' (identifies it as a data warehouse table). The additional 2 characters or digits 'XX' indicate the general class of information the table belongs to. They also serve to group and sub-group tables so that they organize themselves in a list of tables when sorted alphanumerically. This approach allows the

data warehouse tables to be easily distinguished from any source and staging tables.

There are examples of tables which have the following format:

DXX_AAAAAAA_STD (e.g. DAF_COSTCODE_STD: Cost code standard)

These are examples of corporate 'standards' tables. A list of table prefixes appears below.

Table Prefixes

P1	P2-P4	Position 5-end – Description of Prefix
D	XX	Leading prefix identifies this table as belonging to a data warehouse or data mart.
D	00_	This code is used exclusively for the license table.
D	AF_	Authority or lookup tables. These tables may be used as outrigger dimensions (eg. Calendar, Address Book, Data Source, etc.). May require views or need to be aliased to work in the data warehouse context.
D	AP_	Accounts payable dimensions.
D	AR_	Accounts receivable dimensions.
D	AS_	Asset dimensions
D	DW_	Drawing/document dimensions
D	EQ_	Equipment and Parts dimensions
D	FA_	Fact table
D	GL_	General Ledger (COA)
D	LA_	Labor (including payroll) dimensions
D	OR_	Order and Contract Dimensions
D	PL_	Planning Dimensions
D	PJ_	Project Dimensions
D	SC_	Schedule Dimensions
D	SY_	System – Administration Entities

FIELD NAMING

All fields follow the same bi-camel naming convention:

zzzzBbbbbCccccDdddd (e.g. afccCostTypeId: Cost Type Id)

The first set of characters ('zzzz') are a prefix code that links the field to a specific table. The 'afcc' code in the example indicates the 'Cost Code Type Id originates in the 'Cost Code Authority File'. An 's' or 'a' is added to the first 4 characters when the table is a standard table (_STD: zzzzsBbbbbCcccc) or an alternate table (_ALT: zzzzaBbbbbCcccc). With a few exceptions, the extended name is used for clarity. A few abbreviations are used however. For example, 'Desc' is used as an abbreviation for description. Standard abbreviations are listed in the table below.

Standard Abbreviations

Abbrev.	Full Name
Desc	Description
Doc	Document
CC	Construction Coordinator
Class	Classification
Eng	Engineering
Id	Identifier
Jur	Jurisdiction
No	Number
Qty	Quantity
Std	Standard
Uom	Unit Of Measure

A suffix is used to identify primary keys, foreign keys, natural keys, candidate keys and aggregates. Note that there are redundant values in the domain tables (relative to the fact table). These fields are useful in the development environment but can be eliminated in the test and production environments.

P1	Example	Description

_PK	pjpr_PK	Project primary key (from the project table). Note that all primary keys are surrogate type keys. And there is only one per table.
_FK	afcoEmployer_FK	Company foreign key. Note that all foreign keys are surrogate type keys.
_NK	afccCostCodeId_NK	Cost Code identifier – Natural Key. Unique key or set of keys for a record. Should be used in combination with 'Source' in a 'multi-source' data environment.
_CK	afcoTin_CK	Company Identifier. Alternate key.
_AG	pjtaBudget_AG	Identifies a field as an aggregate value in a dimension table. This may be an aggregate from within a record (e.g. Total = (Qty*Unit Rate)+Tax+Freight) or it may be an aggregate from another table (e.g. the sum of an order's detail lines in an order 'header').

SPECIAL CASE

Stubs have been added to the warehouse for tables that don't exist. These 'stubs' have a prefix of 'xxxx' indicating that there isn't an association with a table. Example:

DAF_ADDRESSBOOK	xxxxDivision_FK
	afbkDivisionId
	xxxxDepartment_FK
	afbkDepartmentId
	xxxxOffice_FK
	afbkOfficeId

Note that the associated identifiers follow the format described above and are present even though if populated will be EMPTY or NULL.

CHANGE MANAGEMENT

The goal of change management will be to make sure interested users and developers understand how the model has changed from version to version. A product management process will be put in place to identify requested changes, report change status, and implementation schedules.

Initially this will consist of a change request and a change/release logs managed by ASA. Change requests from the community and ASA will be logged and available in a shared space.

As a PCDW community develops, a committee will be formed to oversee the change management process. As with most data warehouse exercises, 'finished' is a relative concept.

DEVELOPMENT, TEST, and PRODUCTION

The PCDW is delivered as a development environment. It has native key representations (identifiers) in the dimensions and in the fact table. It also has values (aggregates) in the dimensions and some redundant values. It is valuable to have such features in the development environment but you may want to remove some of these elements from the Test and Production environments.

RELATIONSHIPS (BETWEEN TABLES)

Parent-Child (one to many) relationships between tables are identified by the field naming conventions. Relationships are identified within the model but are not built with the model because relationships add considerable overhead to the load portion of the ETL process. A brief discussion of the approach to Primary and foreign key relationships appears below.

Primary keys (parent table - _PK suffix) and foreign keys (child table - _FK suffix) are both designed to be surrogate keys (integer). Because there are no multi-key primary keys, the attribute's 4 character field prefix is sufficient to identify a foreign key's origin (table). For example:

DAF_COSTCODE (parent) afcc_PK --< afccCostCode_FK (child) DPJ_PROJECT

For outrigger tables (eg. calendar, address book, company, etc.) a context appropriate field name will be used in conjunction with the prefix and _FK suffix to distinguish it from other foreign key instances of the same type in the child table. For example, the calendar authority file (DAF_CALENDAR) is referenced multiple times in the Task table. Because of this, a single key, afcaDate_FK is not sufficient to cover all calendar instances and a context name is applied.

DAF_CALENDAR (parent) afca_PK --< afcaActualStart_FK (child) DPJ_TASK
afcaActualEnd_FK (child) DPJ_TASK
afcaContractStart_FK (child) DPJ_TASK
afcaContractEnd_FK (child) DPJ_TASK

Natural Keys: Natural keys are identified with a _NK suffix. To be unique, most natural keys have to be combined with a source foreign key (surrogate) to be unique in a multi-source environment.

SYSTEM OF RECORD - SOURCE DATA

The PCDW relies heavily on an organization's accounting data as its system of record (SOR). The reason for this is twofold: first, accounting data drives an alignment between accounting and the business (projects and operations) for both process and data (if an order doesn't exist in the accounting system, it doesn't actually exist), and second, data from accounting systems is usually clean due to strict business rules enforced by the accounting system software. This should translate into less cleanup and fewer integrity problems in the ETL process. This may also make cleanup of other source data easier if the data warehouse becomes an authority for select data (project id, work classifications, etc.) and if this data is used to check incoming data from other sources (schedules, project management systems, etc.)

One question is sure to come up; What if the data doesn't exist in your accounting system? The PCDW is built 'wide' to accommodate a lot of different types of information. It would actually be surprising if you had data to fill all the fields in the tables. All accounting systems have or should have purchase orders (orders), accounts payable, accounts receivable (billing), payroll systems, and supporting information like supplier (aka: vendor, contractor, etc.), address books, etc. You will find that most of this information can be fit to the tables provided. The tables have been named to make it easy to find matching data sets.

If you don't have a Job Cost accounting system you likely won't have select project information directly associated with your accounting information. If you look at the PCDW's structure, one thing you will soon realize is that project ids, cost codes, WBS codes, etc. are just a set of 'tags' that are hung on each transaction. If project information is not part of your accounting system or process, they can be attached by ancillary software that would allow you to create and assign project identifiers to transactions. From a process point of view, this might even be easier than assigning projects to transactions as part the regular business processes associated with accounting.

DATA SILOS

A good bit of data in the PCDW can and should originate from accounting systems. There are a number of accounting silos that may not have much visibility between them. The PCDW brings much of that

information together and makes it visible in project and other contexts.

CONFIDENTIAL DATA

There are areas of sensitive data which will vary from organization to organization. Employee and contractor pay rates are typical examples. These tables may be protected by establishing specialized schemas and restricting access. In addition special tables have been provided (e.g. DLA_TIMECARDSUMMARY) to aggregate and obfuscate rate information. For example, the 'effects' of the rates can be masked to some degree in the fact table by aggregating labor cost by craft or some other dimension.

ETL - LOAD ORDER

In general, the load order is from the outside-in: Authority, Dimension (header), Dimension (detail) and fact. Parallel loading is possible for a single source and for multiple accounting sources. Non-Accounting sources with project or other 'authority' dependencies should be loaded last.

CAPACTIY PLANNING

Capacity requirements will vary based on the way an organization uses the system. However, the PCDW is similar to a financial data warehouse. To give you a sense of scale, one of these warehouses with an average of $25M (US) per year consumed 500GB over a 4 year period.

APPENDIX A: PROJECT CONTROL DATA WAREHOUSE FEATURES

PCDW MISSION: Answer the question, "How is the project going?" from many points of view.
DISTRIBUTION: Global
INDUSTRY(s) SERVED: All (All organizations have projects)
FUCTIONAL AREA: Project Control
SUPPORT: By Contract, On-Line Community, On-line Documentation.

PCDW OS PLATFORMS: Constrained by the database selection
PCDW DATABASES: Microsoft SQL (2000 through recent), Oracle, DB2, Access, MySQL, and other SQL92 Compliant databases.
PCDW ACCESS: ODBC, OLE, or Native Access drivers.
PCDW Structure: Dimension-Fact. Built using relational modeling tools
PCDW OLAP SUPPORT: Not Included. Owner's choice.
PCDW BI: Not Included. Owner's choice. May be constrained by the database selection
ETL: Not Included. Owner's choice. Dependent on source(s)
STAGING TABLES: Not Included (part of the ETL process).

PCDW ARCHICTURE: Snowflake
PCDW TIME PERSPECTIVE: Current Data Summaries
PCDW UPDATE RATE: Recommend Work Shift or Daily for organizations with a constant stream of work.
REQUIRED DATA SOURCE(S): Accounting Systems (ideally Job Cost or systems with project controls).
OPTIONAL DATA SOURCE(S): Scheduling, Project Management, Time Keeping.
DATA RESOURCE: Consider using as an organizational resource for select data (Project List, Standard Cost Codes, Facility List, etc.)
PCDW GRAIN: Transaction Level in the Fact table
PCDW SECURITY ISSUES: Payroll and Contract transactions in the fact table
PCDW SLOWLY CHANGING DIMENSIONS: Rare (see time perspective)
PCDW HIERARCHIES: Several Ragged and Fixed (Project, Cost Code, WBS, etc.)

LANGUAGE: English (will add additional languages based on community interest)
DATABASE NAME: Organization's choice (e.g. PCDW20140528)
LICENSE: License table identifying ownership (required)
TABLE NOTATION: General table type code plus full name
FIELD NOTATION: Bi-Camel verbose with table association prefix and suffix attachments for primary, candidate, and foreign keys.

DEVELOPMENT and PRODUCTION: Accommodation made for development and production versions of the Warehouse.
DATA MARTS: Not Included. Can be developed.
DATA FEDERATION: Source table included to allow consolidation of data from multiple sites.
SNAPSHOTS: Select snapshots available beginning with v1.02

CHANGE CONTROL: Formal change control model to be determined. Will use change logs in the interim.

- Initial version was identified as 0.0n with changes identified as incremental changes 'n' until the first production release.
- First production release designated as v1.00. Incremental changes identified as '1.0n'.

PCDW DATA INDEXES: Primary Key only.*
PCDW DATA STORED PROCEEDURES: None*
PCDW DATA VIEWS: None*
PCDW DATA ALIAS: None*
* Nature and structure to be determined by implementers.

