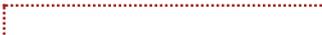


# *Data Architecture Document Template*



*Version 2.0*

# Authors

| [Name]                                       | [Name]                                       | [Name]                                       |
|----------------------------------------------|----------------------------------------------|----------------------------------------------|
| [Company].<br>[Company Address]<br>E-mail ID | [Company].<br>[Company Address]<br>E-mail ID | [Company].<br>[Company Address]<br>E-mail ID |

# *Version history*

|                    |                | <b>Draft</b>      |               | <b>Final</b>         |                 |
|--------------------|----------------|-------------------|---------------|----------------------|-----------------|
| <i>Description</i> | <i>Version</i> | <i>Draft Date</i> | <i>Author</i> | <i>Approval Date</i> | <i>Approver</i> |
|                    |                |                   |               |                      |                 |
|                    |                |                   |               |                      |                 |
|                    |                |                   |               |                      |                 |
|                    |                |                   |               |                      |                 |

# Table of contents

---

|                                                                   |    |
|-------------------------------------------------------------------|----|
| Data Architecture Document Template                               | 1  |
| Authors                                                           | 2  |
| Version history                                                   | 3  |
| 1. Introduction                                                   | 7  |
| <hr/>                                                             |    |
| 1.1. Purpose                                                      | 7  |
| 1.2. Approach                                                     | 7  |
| 1.3. Document organisation                                        | 7  |
| 1.4. Intended audience                                            | 7  |
| <hr/>                                                             |    |
| 2. Data model overview                                            | 8  |
| <hr/>                                                             |    |
| 2.1. Data architecture strategy                                   | 8  |
| 2.2. Conceptual data model                                        | 8  |
| 2.3. Enhanced/Modified data stores                                | 8  |
| 2.4. Leveraged data stores                                        | 8  |
| 2.5. Data access                                                  | 8  |
| 2.6. Data dictionary                                              | 9  |
| 2.7. Assumptions, dependencies and additional constraints         | 9  |
| <hr/>                                                             |    |
| 3. Detailed data model                                            | 10 |
| <hr/>                                                             |    |
| 3.1. Component-1 (Duplicate this section for each data component) | 10 |
| 3.1.1. Entity-relationship diagram                                | 10 |
| 3.1.2. Tables                                                     | 10 |
| <hr/>                                                             |    |
| 3.1.2.1. Table 1 (Duplicate this Section for Each Table)          | 10 |
| 3.1.3. Indexes                                                    | 10 |
| 3.1.4. Triggers                                                   | 10 |
| 3.1.5. Stored procedures                                          | 11 |
| 3.1.6. Security                                                   | 11 |
| 3.1.7. Performance                                                | 11 |
| 3.1.8. Capacity                                                   | 11 |
| 3.1.9. Data access                                                | 11 |
| 3.1.10. Error handling                                            | 11 |
| 3.1.11. Installation and deployment strategy                      | 11 |
| 3.1.12. Data initialisation                                       | 11 |
| <hr/>                                                             |    |
| 3.1.12.1. Cleansing                                               | 12 |
| 3.1.12.2. Conversion                                              | 12 |
| 3.1.12.3. Migration                                               | 12 |
| 3.1.13. Data management                                           | 12 |
| 3.1.14. Documentation and training impacts                        | 12 |

|                                              |    |
|----------------------------------------------|----|
| 3.1.15. Unit test cases                      | 12 |
| <hr/>                                        |    |
| 3.2. Component-2...                          | 12 |
| <hr/>                                        |    |
| 4. Software blueprint                        | 13 |
| 5. Additional information                    | 15 |
| <hr/>                                        |    |
| 5.1. Acronyms, abbreviations and definitions | 15 |
| 5.2. Open issues and future considerations   | 15 |
| 5.3. References and related documents        | 15 |
| <hr/>                                        |    |
| 6. Appendices                                | 16 |
| <hr/>                                        |    |
| 6.1. Appendix A – Data dictionary            | 16 |
| 6.2. Appendix B – Software blueprint         | 17 |

<ITALICS are special instructions for completing the template. When creating a template, insert detailed instructions in italics. Before finalising a deliverable, **DELETE all italic text**, instructions and unused material.

- On Title Page: Insert document name and date of submission;
- On Author and Version History Page:
  - Author Table: Insert author names and delete generic references;
  - Version History Table: track ongoing history of changes to this document. Approver should be highest level person, persons or committee required to validate and adopt this document;
  - Copyright Version: May require adjustment based on the Terms and Conditions of the Engagement Contract. Year reference may require adjustment based on current date;
  - Select View>>Header and Footer to adjust Headers/Footers. Select item, right click and select Update Field.
- On the Table of Contents Page: Right Click within the Table of Contents. Select Update Field>>Update Entire Table;
- This document contains generic text and formatting references to assist in development. Adjust, duplicate by adding additional sections, build on and replace this text as appropriate for your document;
- When using this template to create a Technique, the following applies:
  - Reference the Activity(ies) or Deliverable(s) the Technique applies to;
  - Describe parameters of use (for example, only applies to federal agencies, only applies to specific technology);
  - Describe the advantages of following the technique's leading practice (for example, time savings, skills required, quality).
- Verify PwC owns the copyright on any text or graphics insert into this template. Insertion of Client material or logo requires the Client's written permission. If ownership or permission cannot be verified, do not use.>

# 1. Introduction

*<Write a brief introduction describing what the reader will find in the subsections of section 1.>*

This document focused upon the ...

## 1.1. Purpose

*<Defines the business definition/scope of the data model.>*

This...

## 1.2. Approach

*<Describe how the data model was determined: who, what, where, when, and how? If there were other documents used it could be important to have precise references to them (i.e. date, version, etc.). If it's based upon discussions/meetings it would be appropriate to list the meeting information (i.e. attendees, dates) if there are no meeting minutes that can be referenced. Also provide a brief summary of any the key drivers.*

*Consistent with the engagement's quality practices provide traceability to requirements. Often the data model has a supporting relationship to the business/functional and/or technical (non-functional requirement). Describe this supporting relationship.>*

This...

## 1.3. Document organisation

*<The section provides a brief overview of how the document is organised.>*

This...

## 1.4. Intended audience

*<This section explains to whom the document is directed and which sections may be most useful for the corresponding personnel.>*

## **2. Data model overview**

*<This section describes why this data model is required (i.e. what features are driving the need), and the overall feature/capability that the database(s) is a part of (relationship to components). It should mention if data is shared with other systems/components/applications.>*

*The scope is all data within the solution including operational/configuration, metadata and logs, etc. that may be in flat files. These should be described for completeness and also because with a distributed environment often central servers are used for such data as well as directory services, push/pull, and syncing techniques. Understanding how it will be managed and accurately maintained is important.*

*Applications that may be part of the broad solution should have their own data model. The information that is captured here should compliment the other data model documents (not repeat them) and be sufficient to explain the implementation. The detail sections should support any development efforts.*

*If there are multiple components (databases) in this solution provide a diagram here showing the relationships of the various components. Also provide a description of how the components work together via use cases or other means of describing the concept of operation. This overview should be sufficient to provide a transition and introduction to the more detailed information that follows.>*

**This...**

### **2.1. Data architecture strategy**

*<Describe here the principles used in designing the data model: what were the important criteria and what is being done to meet that criteria within the design (e.g. minimise maintenance, support certain volumes, completely leverage a legacy DB without modification, replace a legacy DB, combine several existing DBs, reduce data redundancy, identifiable improvements from the embedded environment)? This may be a table or bullet list with the criteria described and then the strategy for meeting it. Basically this should provide the rationale for the overall design.>*

*Describe the constraints, tradeoffs, alternatives that were considered, major decisions/selections made and the rationale for the decision/selection.>*

### **2.2. Conceptual data model**

*<This section describes the data model from a business point of view. A graphic model showing the main relationships (including cardinality) between the fundamental elements in the solution should be included. Constraints in the model also should be established. Large-scale data-centric software may require the definition of a dedicated logical data model. If such a model already exists that document should be referenced.>*

### **2.3. Enhanced/Modified data stores**

*<Provide a list and a brief introduction of any modified data stores (databases) that are within the architecture – why the DB is being modified, etc.>*

### **2.4. Leveraged data stores**

*<Provide a list and a brief introduction of any data stores (databases) that are within the architecture – that exist and will not be modified.>*

### **2.5. Data access**

*<Describe any common method, approach, data abstraction, object hierarchy, or class structure that will be developed/used for programmatic data access. If no such mechanisms will be used say so and describe the*

*basic data access approach.. If the details of this are in a detailed design document refer the reader to the appropriate document.>*

## **2.6. Data dictionary**

*<Provide an overview of the data dictionary and reference the data documents being used as input. Describe how the dictionary compliments this document and should be used with this document. If there is no data dictionary or the dictionary is limited and included in Attachment A provide the rational for having done so.>*

## **2.7. Assumptions, dependencies and additional constraints**

*<Describe here any assumptions, dependencies or additional constraints upon which the accuracy of this document relies on. Especially important would be to describe any client plans/projects in progress that will have some impact/influence on the data model.>*

This ...

# 3. Detailed data model

## 3.1. Component-1 (Duplicate this section for each data component)

<Provide a brief overview of the data component (database), what its data domain is purpose in the architecture, scope, etc. Include in the description what the relationship of this component has with others. Use cases or other techniques to describe the concept of operation should be used in the discussion.>

### 3.1.1. Entity-relationship diagram

<Place an ER diagram here.>

### 3.1.2. Tables

#### 3.1.2.1. Table 1 (Duplicate this Section for Each Table)

<For each table in the database there needs to be a detailed design describing its contents, columns, and relationship to other tables. Should information be migrated from another application, the Source column identifies the source. For databases that are part of an installed application, only those tables related to data that will be used in the solution should be listed, and only columns containing information used in the solution need to be included.>

**Table 1:Description**

| Column. Name | Description | Data Type                       | Constraints | Mandatory | Unique | Source |
|--------------|-------------|---------------------------------|-------------|-----------|--------|--------|
|              |             | Number (10), Varchar2 (20) etc. | PK, FK      | yes/no    | yes/no |        |

### 3.1.3. Indexes

<This section lists the indexes defined in the database. If databases are part of an installed application, only those indexes related to share data as part of an integration should be listed.>

#### Indexes

| Index name | Table | Column | Unique |
|------------|-------|--------|--------|
|            |       |        | yes/no |

### 3.1.4. Triggers

<This section describes defined triggers in the database. If databases are part of an installed application, only those triggers handling data as part of the integration should be listed.>

#### Triggers

| Trigger Name | Table | Column | Event | Description |
|--------------|-------|--------|-------|-------------|
|--------------|-------|--------|-------|-------------|

---

### **3.1.5. Stored procedures**

*<This section describes the stored procedures in the database that are used in the solution. If databases are part of an installed application, only those stored procedures handling data as part of the integration should be listed.>*

#### **Stored Procedures**

---

| <b>Store Procedure Name</b> | <b>Arguments</b> | <b>Description</b> |
|-----------------------------|------------------|--------------------|
|-----------------------------|------------------|--------------------|

---

### **3.1.6. Security**

*<This section should describe any impacts to security or vulnerabilities that may exist. It should list any roles defined in the database and their permissions.>*

### **3.1.7. Performance**

*<This section describes any impacts on performance that are anticipated. It includes a description of either design characteristics intended to improve, or minimise the impact on performance.>*

### **3.1.8. Capacity**

*<This section describes any impacts on capacity. It includes a description of either design characteristics intended to improve, or minimise the impact on capacity.>*

### **3.1.9. Data access**

*Describe the method, approach, data abstraction, object hierarchy, or class structure that will be developed/used for programmatic data access. If the details of this are in a detailed design document refer the reader to the appropriate document.>*

### **3.1.10. Error handling**

*<This section needs to describe any error handling built into the component. If this component generates any error messages they should be listed with a description of their likely cause and a potential remedy if appropriate. The error message specification will be needed by testers as well as during the development of operational/user documentation.>*

### **3.1.11. Installation and deployment strategy**

*<This section describes how this database needs to be installed including required software and configuration or setting of environment variables, etc. This section should be included only for new databases developed for the solution. Installation instructions for databases that are part of an already embedded application do not need to be included in this section unless the DB has been modified.>*

### **3.1.12. Data initialisation**

*<This section describes how the data will be initially populated. For example, for new security tables user IDs need to be added to roles and, if a column was added to an existing table it will need to be populated. Many projects involve migration of data, in which cases the sections that follow come into play. If there is a migration planned this section should describe the strategy for doing so as an introduction to the sections that follow.>*

### 3.1.12.1. Cleansing

<Describe any data cleansing that is planned or if none is planned say so>

### 3.1.12.2. Conversion

<Describe any data conversion that will be necessary to migrate the data – may include data types, mapping to completely new values, or adjusting other characteristics (i.e. length, etc.). Describe the tools to be used, processes, etc. If there is no conversion planned state there is none.>

### 3.1.12.3. Migration

<Describe the data migration that will be necessary – if data will be manually entered or typed in prior to going into production describe that as well. Describe the tools to be used, processes, etc.>

### 3.1.13. Data management

<Describe here what data management activities are recommended – backup, defrags, log maintenance, support of disaster recovery, etc.>

### 3.1.14. Documentation and training impacts

<This section should describe what should be included in the documentation and training as a result of this design.>

### 3.1.15. Unit test cases

<In many cases this section may be abbreviated or not applicable as the unit test plan for the DB is performed in conjunction with other functional parts of the solution. However any tests that will be performed to verify that the data store has been accurately configured and functions as designed should be described here.>

<If there are a large number of cases or if this table is to be reused to track test results, consider putting it in Excel or other tool that will allow it to be more productively managed/reused (i.e. sorted, filtered, etc.). See the Testing work stream for applicable templates.>

#### Unit Test Cases

| Test case Number                  | Feature                             | Test Description                    | Inputs                  | Expected Results      | Additional Notes                             |
|-----------------------------------|-------------------------------------|-------------------------------------|-------------------------|-----------------------|----------------------------------------------|
| Use the projects numbering system | Short name for feature or component | Describe the test that will be done | Data that will be input | Data that will result | Any additional criteria – response time etc. |

## 3.2. Component-2...

## 4. Software blueprint

*<The Design Software Rollout (Deployment) detail design deliverables should describe any DB creation or initialisation scripts, etc. If it does, delete this section and Appendix B, otherwise complete Appendix B, and further describe what the blueprint specifies and any dependencies or other details that may help understand the file structure. Think in terms of what someone else would need to know, who perhaps is taking over support of the code base.>*

*<The following section contains possible additional paragraphs you may wish to include in your document. Use any or all of these paragraphs, or if none, then delete the entire section.*

*Any of the information below may be provided in a list in the body of this document, as a reference to an appendix, or as a reference to another document.>*

# **5. Additional information**

## **5.1. Acronyms, abbreviations and definitions**

*<Provide an alphabetical listing of acronyms, abbreviations, terms and definitions needed to understand this document.>*

Body 1 – Verdana, 9 pt.

## **5.2. Open issues and future considerations**

*<If there are known issues, risks or considerations: describe, give timeframe, possible resolution>*

Body 1 – Verdana, 9 pt.

## **5.3. References and related documents**

*<List the title, version/publishing date of referenced documents, websites, or other relevant references. If copyrighted documents are referred to, the copyright information must be appropriately referenced.>*

Body 1 – Verdana, 9 pt.

# 6. Appendices

<Appendices may be used to provide information published separately for convenient document maintenance, such as classified data, or for providing supplemental material. The main body of the document should contain at least one reference to each Appendix. Appendices are listed in alphabetical progression (A,B,C).>

## 6.1. Appendix A – Data dictionary

<A data dictionary can be used to supplement the data model document. It may be especially useful to describe representations of commonly used fields/identifiers that should have a common representation. For example:

- Names: LastName, FirstName;
- Phone numbers: nnnnnnnnnn (rather than (nnn) nnn-nnnn or other representations common in the USA);
- US social security numbers (with or without dashes), etc.
- It can also be used to list the values of a field when only a small subsets of values is permissible such as:
- Status: Open, Closed;
- Where there are multiple representations within the data model (perhaps because of a heritage implementation it is helpful to include notes indicating such relationships).

For efforts with large data requirements/specifications the data dictionary may be substantial and should be a standalone document supporting the development/implementation of several integrated applications.>

| Field or Tag Id | Data Type    | Value Description                       | Reference                                       | Notes                                                                                  |
|-----------------|--------------|-----------------------------------------|-------------------------------------------------|----------------------------------------------------------------------------------------|
| Customer Name   | (ASCII char) | Last, First <30 characters no spaces    | XYZ DB, entered via GUI                         | >30 characters will be truncated                                                       |
| Order Status    | 1 byte ASCII | O, C                                    | Order Status in ABC DB, Ostatus in XYZ DB       | Should be consistently displayed as Open and Closed                                    |
| Order Number    | ASCII        | AAAAA any five alpha numeric characters | ABC DB, originates from order entry application | Four digit numerical representation to be converted when input to 5 digit alphanumeric |

## 6.2. Appendix B – Software blueprint

<Note, the columns of this table should be modified to the software environments structure of subsystems, components, shared libraries, etc.

In the table below: the build unit is the module name as defined in the configuration management and/or software build system. The Deliverable Unit is the file(s) that is delivered to the production environment. Having a complete list of the delivered files is critical for verification so that none are missing from the release packaging. This can be especially critical to partial upgrades where a file being out of date may not be initially noticed. For a given release each detailed design software blueprint should rollup to the blueprint in the Software Release Plan.

If suitable, enhance the table to indicate which files within the build unit are new or changed.>

| No. | Build Unit | New/Changed | Files                      | Deliverable Unit | Subsystem     | Notes                                           |
|-----|------------|-------------|----------------------------|------------------|---------------|-------------------------------------------------|
| 1   | ABCI       | New         | Abci.c,<br>comm.h          | abci.o           | Communication | Installed in<br>Communication<br>Subsystem only |
| 2   | Log        | Changed     | Log.c, print.c<br>format.h | Log.lib          | All           | Shared library<br>common to all<br>subsystems   |
| 3   |            |             |                            |                  |               |                                                 |

