

## Database Document

**USAP,  
Responsibility**  
These tables are static and are loaded from a script

**Evaluation responsibility  
Evaluation results**  
This table contains the information for the current evaluation.

This document describes the data model for APLUS requirements and evaluation. This data model contains two tables that are fixed and loaded from a script (the USAP and Responsibility tables) and two tables that are generated based on the USAPs selected during the requirements step. It does not include the concept of project or users. It also does not include the ability to add options. These options are extensions that could be added later.

1. **USAP table**  
This table contains the details of all the foundational and the end user USAPs.
2. **RESPONSIBILITY table**  
This table contains the details of responsibilities of the Foundational USAP's and the End User USAP's. It stores the hierarchy in the form of a tree
3. **EVALUATION RESPONSIBILITIES table**  
This table stores the responsibilities of a particular evaluation based on USAPs selected. It is generated by replacing the parameters included in the responsibilities in the RESPONSIBILITY table by text representing the particular usaps chosen.
4. **EVALUATION RESULTS table**  
This table contains the results of the current evaluation. It is generated from the EVALUATION RESPONSIBILITIES table by having one row for each responsibility/usap selected pair. The evaluation is performed per responsibility per usap.

The EVALUATION RESPONSIBILITIES table has one row per responsibility and so the recording of the results requires another table.

## Table Structure

### 1. USAP table

This table contains the details of all the foundational and the end user USAPs. The fields are

- USAP\_ID int(5) – Auto number for unique identification of the records (primary key)
- USAP\_Name text– Contains the name of the USAPs both the foundational and the end user USAPs
- Type text – Indicating Foundational or End User USAP
  - Foundational USAP – F
  - End User USAP – E
- Purpose, Glossary, Justification, Scenario, Overview, Assumption, Shared\_Assumption and Usability\_Benefits text – Written from the USAP pattern language document.

### 2. RESPONSIBILITY table

This table contains the details of responsibilities of the Foundational USAP's and the End User USAP's. It stores the hierarchy in the form of a tree.

- Responsibility\_ID int(5)– Unique reference (primary key) for the responsibilities
- USAP\_ID int(5) – Foreign key that references to the ID of the USAP from the USAP table
- Current\_ID text – Stores the ID to identify the current node
- Current\_Name text – Stores the Name of the current node
- Child\_ID text – Stores the ID to identify the child node
- Current\_text text –Stores the text for this responsibility
- Type text – This corresponds to the type of the current node. The type can be of four categories
  - Foundational USAP – F
  - End User USAP – E
  - Category – C

- Responsibility – R
- Rationale, Implementation\_Details \_ text – Taken from the USAP pattern language document

Example of storing data in the Responsibility table

USAP Pattern Language Document (Sample)

## **1.1 Authoring**

### **1.1.1 Create a SPECIFICATION**

1.1.1.1 The system must provide a way for an authorized author to create a SPECIFICATION. (See Authorization Foundational USAP with parameters USER=Author and ACTIVITY=Author the SPECIFICATION.)

#### 1.1.1.1.1.1 Rationale

A SPECIFICATION doesn't exist unless it is created.

#### 1.1.1.1.1.2 Implementing this responsibility

The portion of the system that renders output must render a UI that allows the parameters to be specified and displays existing values.

The portion of the system that accepts input from the user must accept parameters.

There must be a portion of the system with a mechanism to create new SPECIFICATIONS.

1.1.1.2 Consider providing default values for all specifiable parameters when a SPECIFICATION is created. (Providing defaults simplifies the creation process, but may increase the probability of error for environments that deviate from those defaults. If the cost of error is high, then consider not providing defaults or requiring confirmation of each value. Default values can be changed through modification.)

#### 1.1.1.2.1.1.1 Rationale

System has to have something so defaults might be provided.

Circumstances might be very similar so defaults might capture that similarity.

Authors may want to be efficient and defaults may save authoring time.

Authors may only be interested in changing specific aspects of the SPECIFICATION, so having defaults for the rest of it is useful.

#### 1.1.1.2.1.2 Implementing this responsibility

The portion of the system that creates a new SPECIFICATION must assign defaults.

The above data is stored the responsibility table as

Responsibility_ID	USAP_ID	Current_ID	Current_Name	Child_ID	Child_Name	Type	Rationale	Implementation_Details	Additional_Responsibili
20	3	3	Authoring	3.1	Create a SPECIFICATION	f	NULL	NULL	NULL
21	3	3.1	Create a SPECIFICATION	3.1.1	The system must provide a way for an authorized au...	c	A SPECIFICATION doesn't exist unless it is created...	The portion of the system that renders output must...	NULL
22	3	3.1	Create a SPECIFICATION	3.1.1	The system must provide a way for an authorized au...	c	A SPECIFICATION doesn't exist unless it is created...	The portion of the system that renders output must...	NULL

### 3. EVALUATION Responsibilities table

This table stores the details of a particular evaluation pertaining to a user for a corresponding project

- Evaluation\_Responsibility\_ID int(5) – Unique identifier (primary key) that identifies the evaluation\_responsibility. This is a number assigned sequentially
- Type text - This corresponds to the type of the current node. The type can be of four categories
  - Foundational USAP – F
  - End User USAP – E
  - Category – C

- Responsibility – R

- Current\_ID\_FC text – Stores the ID of the current node
- Current\_Name\_FC text - Stores the name of the current node
- Child\_ID\_FC text – Stores the ID of the child node
- Current\_text\_FC text - Stores the text of the responsibility after substitution has been made to reflect the USAPs selected.
- Rationale\_FC text. Stores the rationale for this responsibility after the substitution has been made to reflect the USAPs selected
- Implementation\_Details\_FC text. Stores the implementation details for the current responsibility have the substitution has been made to reflect the USAPs selected.

#### 4. Evaluation\_results Table

This table stores the results of the evaluation. This includes the selected radio button, the discuss check box, and any notes made by the architect.

- Results\_ID int. This is the key for this table and it consists of 1000\*usap code +Evaluation\_Responsibility\_ID.
- Evaluation\_Responsibility\_ID. Int. This is the responsibility ID for this responsibility in the Evaluation\_responsibility table
- USAP\_Code text. This is the code for the USAP stored as usap\_code.
- Checked\_Status int. This represents the status of this responsibility/usap as reflected in the radio button. 1 is “not yet considered”, 2 is “”must modify architecture”, 3 is “architecture addresses this, and 4 is “not applicable”
- Discuss\_status. Int. The status of the discuss check box for this responsibility/usap. 0 is unchecked and 1 is checked.
- Notes text – Stores the contents of the notes field.