Interface Specification

for

<Project>

Version 1.0 draft 1

Prepared by <author>

<organization>

<date created>

Template Version 1.0 approved

*<Change the footer and header text to reflect the correct copyright date, company name, and project name.>*

**Table of Contents**

Revision History ii

1. Introduction 1

1.1 Document Purpose 1

1.2 Product Overview 1

1.3 Operating Environment 1

1.4 References 1

1.5 Assumptions 1

2. Interface Diagrams 2

3. Data Interfaces 2

3.1 Interface ID 1 2

3.1.1 Overview 2

3.1.2 Data Types 2

3.1.3 Interface File Formats 2

3.1.4 Communication Protocol 3

4. Software Interfaces 3

4.1 Interface ID 1 3

4.1.1 Overview 3

4.1.2 Interface Specification 3

4.1.3 Timing Issues 4

4.1.4 Communication Protocol 4

5. Hardware Interfaces 4

5.1 Interface ID 1 4

5.1.1 Overview 4

5.1.2 Connection 4

5.1.3 Data and Control Flow 5

6. User Interfaces 5

Appendix: Issues and TBDs 5

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

*<The Interface Specification describes both the external interfaces between the product and its operating environment and the internal interfaces between product components. Each interface needs to be described in sufficient detail that developers can implement it, integration testers can verify its correct functioning, reviewers can verify its correctness and completeness, maintainers can respect the interface when modifying components, and requirements analysts and system engineers can confirm the ability of the interface to facilitate collaboration among components to satisfy the product’s requirements. As an alternative approach, you may wish to restrict the contents of this specification to only the external interfaces of the product, and to document the interface of each component separately in that component’s detailed design specification.*

*Note: This template contains both guidance text, shown in italics, and* boilerplate text, shown in normal text. *When creating a project document from this template, customize the* boilerplate text *to suit the specific details of the project. Remove the guidance text and insert your own specific information for the project.>*

# Introduction

## Document Purpose

The purpose of this document is to identify and characterize the internal and external interfaces for the <*Product Name*>. The objectives of this interface specification are to facilitate clear and effective communication among the project stakeholders who are responsible for defining, specifying, implementing, testing, using, or conforming to these interfaces. The intended audience for this document include the project’s requirements analyst, system engineer or architect, software and hardware engineers, support and maintenance engineers, testers, marketers, and managers.

## Product Overview

*<Briefly state the name and purpose of the product to which this interface specification applies. Describe the scope within the product of the interfaces to which this interface specification applies.>*

## Operating Environment

*<Briefly summarize the operating environment for this product to help the reader understand the external interfaces. Possible aspects to address include:*

Hardware platform

Platform configuration (e.g., standalone, mainframe, Web server-based workstations on a LAN)

Operating system; include the version of the operating system, as this may determine whether any data conversions are required on the data exchanged through the external interface>

## References

*<List any other documents to which this Interface Specification refers. These may include user interface style guides, contracts, standards, device datasheets, architecture specifications, or a product requirements specification. Provide enough information that the reader could access each reference, such as title, author, version number, date, and URL.>*

## Assumptions

*<State any assumptions that are being made with respect to the product’s internal and external interfaces. Assumptions are statements that are regarded as being true in the absence of definitive knowledge that they are true. Known facts are not assumptions. Assumptions may also need to be documented in the specifications for individual interfaces.>*

# Interface Diagrams

*<Unless already documented elsewhere, such as in an architecture specification, include a block diagram or other graphical representation of the architecture that illustrates the product’s external interfaces and the internal interfaces between components. If this diagram already exists elsewhere, include a hyperlink to that source here. Give each such interface a project-unique identifier and name so it can be referred to in this document and in other project documentation.>*

# Data Interfaces

*<This section specifies both external and internal data interfaces. Data interfaces could be in the form of files or data streams. Alternatively, the data interfaces can be fully specified in the sections of this document labeled Software Interfaces and Hardware Interfaces.>*

## Interface ID 1

*<Instead of actually saying “Interface ID 1”, state the unique identifier and name for the interface. Repeat this section for each discretely labeled data interface.>*

### Overview

* *Description of the data interface and its purpose*
* *Location of the data (internal or external file, database, software component, hardware component, network)*
* *Source and destination for the data (e.g., from product to external hardware device, or from component A to component B)*
* *Assumptions*

### Data Types

* *Name and definition*
* *Encoding (ASCII, Unicode, XML, X12, EBCDIC)*
* *Units*
* *Valid range, including what to do if out-of-range value received and validity of null value*
* *Default or initial value*
* *Invariants, if any*
* *State behavior, including what to do if data is received when component is in the wrong state*
* *Exception handling*
* *Correctness of the data, including accuracy and precision*
* *Timing issues*
	+ *Periodicity*
	+ *Minimum, average, and maximum arrival rate in each relevant component state*
	+ *Maximum time before first input or output is expected, including what to do if data is not received within maximum time before expected first input (timeout)*
	+ *Will data become obsolete after a specific time? What should be done with obsolete data?*

### Interface File Formats

*<If this interface is used to transfer multiple files, provide the information requested below for each file.>*

* *Sending (source) directory, file name, and physical location*
* *Receiving (destination) directory, file name, and physical location*
* *Average and maximum file size*

*<Provide the data element field information indicated below for each record type.>*

* *Record structure and data elements for each record*
	+ *Record number*
	+ *Fixed or variable length*
	+ *Field delimiter (if variable length field)*
	+ *End of record mark*
	+ *Data element name*
	+ *Starting column position (if fixed format)*
	+ *Data type and length of data element*
	+ *Edits to perform to validate data*

### Communication Protocol

*<Describe the mechanisms used for data exchange, such as TCP/IP, FTP, etc.>*

# Software Interfaces

*<This section describes the interfaces between product software components and external software or hardware components, as well as internal software–software and software–hardware component interfaces. Not all of the interface specification elements may apply to every software interface.>*

## Interface ID 1

*<Instead of actually saying “Interface ID 1” state the unique identifier and name for the interface. Repeat this section for each discretely labeled software interface.>*

### Overview

* *Definition of the software interface*
* *Interface purpose*
* *Interface type (e.g., file, data stream, message)*
* *Invariants*
* *Assumptions*

### Interface Specification

* Operation preconditions and postconditions
* *The software or hardware component that is the source of the data element*
* *The software or hardware component(s) that receives the data element*
* *Syntax*
	+ *Parameters (data types, valid values or ranges (provide actual values for constants), accuracy and precision needed, units of measure, validity and meaning of null values, default values, data flow direction)*
	+ *Return value (data type, valid range, validity and meaning of null value, default value)*
	+ *Exceptions thrown*
	+ *Communication protocol (e.g., remote procedure call, remote method invocation, CORBA, DCOM)*
* *Quality attributes for the interface, such as availability, reliability, and performance*
* *For message interfaces, specify the following information:*
	+ *Message name, title, and/or identifier*
	+ *Message purpose*
	+ *Message creation stimulus*
	+ *Communication method*
	+ *Header record information*
	+ *Message data contents, including (if appropriate) record structure and data elements for each record*
	+ *End-of-message indicator or record contents*
	+ *Expected response when the message is received*

### Timing Issues

*<Indicate whether the interfacing components are to execute concurrently or sequentially. If concurrently, describe the method of inter-component synchronization to be used.>*

### Communication Protocol

* *Communication links, bands, frequencies, media, and their characteristics*
* *Message formatting*
* *Flow control, such as sequence numbering and buffer allocation*
* *Data transfer rate, whether periodic or aperiodic, and interval between transfers*
* *Routing, addressing, and naming conventions*
* *Transmission services, including priority and grade*
* *Safety, security, and privacy considerations, such as encryption, user authentication, compartmentalization, and auditing*
* *Characteristics of protocols the system must use for the interface, such as:*
	+ *Priority or layer of the protocol*
	+ *Packeting, including fragmentation and re-assembly, routing, and addressing*
	+ *Legality checks, error control, and recovery procedures*
	+ *Status, identification, and any other reporting features*

# Hardware Interfaces

*<This section describes the interfaces between product hardware components and external hardware components and those between internal hardware components. Not all of the interface specification elements may apply to every hardware interface.>*

## Interface ID 1

*<Instead of actually saying “Interface ID 1” state the unique identifier and name for the interface. Repeat this section for each discretely labeled hardware interface.>*

### Overview

* *Definition of the hardware interface*
* *Interface objectives*
* *Interface type (e.g., cable, optical fiber, physical device)*
* *External hardware component or device*
* *Electrical, mechanical, and functional characteristics of the hardware interface*
* *Assumptions*

### Connection

* *Physical media*
* *Physical configuration (size, shape)*
* *Hardware interface standard*
* *Pin configuration and number*
* *Voltage, frequency, and current*
* *Grounding*
* *Physical compatibility of the interfacing entities, such as dimensions, tolerances, loads, and plug compatibility*

### Data and Control Flow

* *Data rate and robustness*
* *Input data name, type, and description*
* *Output (returned) data name, type, description, and meaning*
* *Control signal name, type, and description*

# User Interfaces

*<If desired, you may include information about external user interfaces in this document. Alternatively, a separate user interface design document could contain that information. The essential elements to include for user interfaces are:*

Standards and conventions to follow for the user interface, such as a product family user interface style guide, screen labeling conventions, text font and style conventions, function and shortcut key assignments, color schemes, branding, and specifications for standard types of dialog elements or controls, such as comment boxes, forms, error messages, and so forth.

User interface architecture, showing the major dialog elements (screens, pages, forms, menus, dialog boxes, line prompts) and the permitted navigation connections between them. A dialog map is a specialized type of state-transition diagram that works well for depicting user interface architecture (Wiegers, Karl E. Software Requirements, 2nd Edition, Microsoft Press, 2003).

Screen layouts for individual user interface displays, including field positions and characteristics, allowed values for fields, values for drop-down lists, defaults, validations to perform on entered data, inter-field relationships, and so forth.>

# Appendix: Issues and TBDs

*<List all major open issues and items remaining to be resolved (TBD = To Be Determined). For each, indicate the target date for resolution and the individual responsible for resolving it.>*

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Target Date** | **Responsible** | **Status** |
|  |  |  |  |
|  |  |  |  |