**Platform-i MSN**

**Requirements Specifications**

**Functional Requirements Specification**

This remark is hidden text and will not be printed:

The items above and items in the headers and footers of this document contain bookmarks for automatic updating. Do not delete these bookmarks or update them manually. Use the UPDATE TOOL instead.

**TABLE OF CONTENTS**

Document change history 3

1 Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Open points 3

1.4 Requirements identification 3

1.5 Document maintenance 3

1.6 Definitions, acronyms & abbreviations 3

1.7 References 4

2 General description 4

2.1 Product perspective 4

2.2 Product(s) 4

2.2.1 MSN messenger xlet 4

2.3 User characteristics 4

2.4 Assumptions & constraints 5

3 Functional requirements 5

3.1 End user functions 5

3.1.1 Presence 5

3.1.2 Messaging 5

3.1.3 Miscellaneous 6

4 External interface requirements 6

4.1 Physical devices 6

4.2 Communication interfaces 6

4.3 Data requirements 7

Document change history

**Date Person Version Reason**

2004-02-24 René Nieuwenhuis 0.1 Initial version

2004-03-19 René Nieuwenhuis 1.0 Approved

GENERAL: The blue coloured text style “explanation text” contains instructions when using the template. The user is free to either adapt the font of the style to become “hidden” or to remove all explanatory text when the document is completed. For more information see the “template” pages on the Intranet.  
PREFACE: An FRS (Functional Requirements Specification) is a document containing a complete description of a product’s external behaviour (that is, with the entire interface between the product and its environment, including other hardware, software, communication ports, and human users) as a statement of what the system will do without defining how it works internally.

The FRS contains two types of requirements: functional and non-functional.

The functional requirements define what the system does with a complete description of all the inputs and outputs to and from the system as well as information concerning how the inputs and outputs interrelate (chapter 0, 3, 4 and **Error! Reference source not found.**).

The non-functional requirements describe the quality and performance of the system (chapter **Error! Reference source not found.**).

The minimum outline that all FRS documents must use is:

1. Introduction
2. General Description
3. Functional Requirements
4. External Interface Requirements
5. User Interface Specification

Given the purpose of the FRS, it is clear that Design Descriptions do not belong in an FRS.

If the product to be defined is a software library, it probably is wiser to use the content of the Software Component Specification (SCS) template instead of the content of the FRS template. For more information, see Work Instruction Architecture & Design.

Attributes of a well-written FRS.

A well-written requirement applies the following criteria:

* Clear & concise It is written in clear, simple and concise language. The primary readers of the document are able to understand the description.
* Testable & measurable A finite cost procedure exists to test the requirement.
* Unambiguous It has only one interpretation.
* Necessary It does not describe superfluous information or explanation.
* Identifiable It has a unique label or a number.
* Feasible It is believed to be realisable.
* Design independent It does not describe software or hardware design decisions.

A well-written FRS contains a set of well-written requirements and applies the following criteria:

* Complete Everything the system is supposed to do is in the document.
* Consistent The requirements do not contradict.

How to organise an FRS

The chapters of this template are mandatory. The sections within the chapters can be chosen differently depending on the project and the specific product.

For more information, read the Work Instruction Requirements Management.

# Introduction

## Purpose

The purpose of the FRS is to get a common understanding between the customer and PDSL on the product requirements of the Platform-i MSN application. The document serves as a basis for the Architectural Design of the Platform-i MSN application.

## Scope

The intended audience for this document are the customer and the development team.

## Open points

Give a list of issues that have not been solved in this document yet. This section should be empty when the document is accepted.

User Interface.

## Requirements identification

Describe how the individual requirements will be labelled and/or numbered in this document. Preferably use a hierarchical format, e.g. <label>.<label>.<number>.

The requirements will be in the format: <label><number>

## Document maintenance

Describe how this document is maintained and how the requirements identification will be handled over time. Typically describe how removal or addition of requirements is handled (e.g. numbers are never renumbered after removal; requirements can only be added at the end of a section with the next higher number).

General rules for document maintenance are described in [WI\_REQ]. Do not repeat these here.

Additional requirements can only be added at the end of a section with the next higher number.

If a requirement is removed, the number remains empty.

## Definitions, acronyms & abbreviations

A list of all definitions, abbreviations and terms used in this document, together with a short explanation.

**Definitions:**

Instant Messenger An instant messenger is an application, which allows instant text communication between two or more people through a network such as the Internet.

Status Predefined presence identifier. With setting the status the user can let others know if he is actively using your PC (or TV in our case). In MSN Messenger you have the following statuses: Offline, Online, Busy, Idle, Be Right Back, Away, On the Phone and Out to Lunch.

Blocking When you block someone, you prevent that person from seeing your status (you always appear offline) and sending you messages.

Buddy Friend, family member, co-worker or other person who is manually added to your buddy list.

Buddy list List of buddies with whom the user can communicate in real time.

Contact list See buddy list.

Passport .NET Passport is an online service that makes it possible for you to use your e-mail address and a single password to sign in to any .NET Passport-participating Web site or service. One of these services is the MSN Messenger.

**Acronyms and abbreviations:**

IM Instant Messenger

MHP Multimedia Home Platform

MSN Microsoft Network

## References

Include essential references here. It is not necessary to include details like author, version and date for all references. E.g. text below can be used:

[REF1] Commercial Requirements Specification of “<project name>“ project  
<Author of the document>  
Reference number <DSE-ygxxxxINI CxSx>  
Version <x>, date <200x-xx-xx>

# General description

## Product perspective

* If the product is independent and totally self-contained, it should be stated here.  
  Otherwise, if the FRS defines a product that is a component of a larger system or project, then describe the relation and identify the interfaces between the component and the larger system or project.
* Describe here all external connections to the product in terms of connectors, hardware and software. The use of a context diagram is allowed if the technique is clear to the customer. Also identify the protocol interfaces (for more information, refer to section 4.2) and the data interfaces (refer to section 4.3).
* If the FRS is part of a product line, refer to the product roadmap and identify the relation between the product and the product roadmap.

The MSN messenger xlet is a MHP version of the popular PC application. It is an application to demonstrate the possibilities of MHP and Platform-i.

The Platform-i MSN messenger application is independent of other projects.

## Product(s)

Identify the product(s) to be produced by name.

In case of more than one product, define how in the rest of the document the diversity between the products is described.

For each product, add the following section:

The produced product will be an MHP MSN messenger xlet.

### MSN messenger xlet

Explain shortly what this product will do. Give a summary of the functions that this product will perform.

Also describe explicitly what the product will not do (but do not add an exhaustive list; restrict the list to the non-trivial aspects).

With the xlet the user shall be able to see online friends, chat with them and see which TV program they are watching. Unlike the PC variant, the xlet cannot transfer files and doesn’t have webcam support.

## User characteristics

Describe those general characteristics of the users of the product that account for the functional requirements. Describe per type of user the context of use in terms of the privileges and the tasks. Typical examples of users are: end users, operators, factory users, service users, and dealers.

Also, non-human users that interface with the product can be included here. Typically, however, non-human usage is a communication interface, see 4.2 for more information.

There are only end-users.

## Assumptions & constraints

Provide a short description of external factors that affect the requirements in the FRS (assumptions) or that will limit the options for designing the system (constraints).

An example of an assumption is that a specific chip will be available. If, in fact, this chip is not available, then this FRS would have to change accordingly.

Examples of constraints are: the operating system to use, interfaces to other applications, safety considerations, regularity policies, higher-order language requirements. Keep in mind that only the non-trivial constraints should be mentioned here.

If we cannot use the MSN messenger service protocol this FRS has to be adjusted to reflect the new situation.

# Functional requirements

For each function (or feature) category, or if appropriate for each individual function, define the functional (behaviour) requirements by means of a short introduction of the function, followed by a definition of inputs and associated outputs (results, effects) on that function.

The User-interface specific aspects of the FRS are not included in this chapter (see chapter **Error! Reference source not found.**). E.g. for a TV set, define here the characteristics of volume control. The fact that the volume control button is present both on the TV-set and on the remote control is defined in chapter **Error! Reference source not found.**).

This chapter will have a section layout that is specific for the product.

It is advised however to at least use the following section layout to make a clear distinction between the types of user (e.g. end user, factory, service user).

## End user functions

Define in function categories the typical functions that are available to the end user.

An example layout could be:

3.1.1 <Function category 1>

3.1.2 <Function category 2>

### Presence

**P1.0 Use of Passport to login to the messenger**

**P1.1** With the use of an existing Passport account the user can login to the messenger.

**P1.2** It is not possible to create a new Passport account.

**P2.0 Maintenance of users own status**

**P2.1** Ability to change and maintain the status which is visible to the buddies. The application provides a list where the end-user can choose the new status.

**P3.0 See the presence status of your buddies**

**P3.1** The application will display the status of the end-users buddies.

**P3.2** When a buddy changes his status, the application shall update it.

**P4.0 Own Nickname**

**P4.1** The end-user is able to change the nickname with which the user is visible to the buddies.

**P5.0 Nickname of buddies**

**P5.1** Show the nicknames of the end-users buddies on the buddy list.

**P5.2** The application updates the displayed nick in the contact list when a buddy changes the nickname.

**P6.0 Appoint an user-defined nickname**

**P6.1** The application provides the possibility to assign a nickname to a specified buddy; this nickname has preference above the nickname set by the buddy itself.

**P7.0 Add buddies to your contact list**

**P7.1** The application provides an option to add a buddy to the buddy list.

**P8.0 Delete buddies from your contact list**

**P8.1** The application has an option to delete a buddy from the buddy list.

**P9.0 Blocking a buddy**

**P9.1** The application has an option to block certain buddies.

### Messaging

**M1.0 Incoming message notification**

**M1.1** The application has a visible notification when an incoming message is received.

**M2.0 Read incoming messages**

**M2.1** Show incoming messages on the screen.

**M3.0 Write a message**

**M3.1** Possibility to create a message and send it to a buddy.

**M4.0 Emoticons in messages**

**M4.1** Show emoticons in incoming and outgoing messages.

**M4.2** Display a list of emoticons to choose from when the user wants to include an emoticon in the message he is writing.

**M5.0 Show (session) history**

**M5.1** Show the history of messages sent and received in the current chat session.

**M6.0 Group conversation**

**M6.1** Chat with multiple buddies together.

### Miscellaneous

**D1.0 TV program**

**D1.1** Ability to request the TV program / channel to which the end-users online buddy is watching at the moment.

**D2.0 New mail notification**

**D2.1** Show a notification when there is new e-mail at the users hotmail inbox.

**D3.0 Check your hotmail account**

**D3.1** Display the inbox of the users Hotmail account

**D3.2** Show e-mails from the users Hotmail account on the screen.

**D4.0 Play games**

**D4.1** Play games with online buddies.

# External interface requirements

## Physical devices

Define the actual input and output devices of the product, typically represented by a picture of the layout of the device.

Examples of input devices are: remote control, mouse, and local keypad.

Examples of output devices are: LCD, TV screen, LED.

Input device: Remote control and maybe a wireless keyboard.

Output device: TV screen.

## Communication interfaces

Describe the external protocols that are used in connecting the product to its environment (see also 2.1). For standard protocols, use a reference and describe what is supported of the standard. For proprietary protocols, define the protocol itself. (Often, the protocol definition is described in a separate document; in that case include a reference here.)

The used protocol for communication is the .NET messenger service protocol. The used version of this protocol will be MSNPv8 which is the standard protocol at this moment.

## Data requirements

Describe the external data that is interfacing to the system. This can be a database or a data stream format. (Often, the data description is defined in a separate document; in that case include a reference here.)

Not applicable.

* Non-functional requirements are typically requirements in the area of e.g. performance, reliability and maintainability. They often have a system wide character and cannot be attached to a specific functional requirement. (This also holds for section 2.4.) Make sure that the requirements are measurable; e.g. “maintainability is important” is not a requirement.
* The following list defines possibly relevant product quality attributes:
* This list is to be used as a checklist to derive the non-functional requirements.
* Preferably, do not use the complete list of attributes as a table of contents. Do not extensively discuss or incorporate those attributes that are regarded less relevant for the application.
* Note that some product quality attributes might result in functional requirements (e.g. security can be realised with a password mechanism, usability is often implicitly included in section **Error! Reference source not found.**, time-behaviour requirements can be addressed specifically per functional requirement). These are typically described in the other chapters of this document. In that case, a reference from this chapter to the other chapters can be useful, to understand how the quality attribute is achieved with functional requirements.