

---

**Egosoft**

---

**Network Integration  
Software Requirements Specification  
For X4**

**Version 0.95D**

Network Integration	Version: 0.95D
Software Requirements Specification	Date: 04/02/08
X4-NET-SRS	

## Revision History

Date	Version	Description	Author
20/11/07	0.9	Initial draft.	Stefan Hett
04/02/08	0.95	Some minor corrections.	Stefan Hett
04/02/08	0.95D	Adjusted the version for the diploma thesis.	Stefan Hett

Network Integration	Version: 0.95D
Software Requirements Specification	Date: 04/02/08
X4-NET-SRS	

## Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms, and Abbreviations	4
1.4	References	4
1.5	Overview	4
2.	Vision	5
2.1	Introduction	5
2.2	Positioning	5
2.3	Project goal and scope	5
2.4	User Descriptions	6
3.	Overall Description	6
4.	Use-Case Model	7
5.	Requirements	8
5.1	Usability	8
5.1.1	Game Engine Developers	8
5.1.2	Mission Designers	8
5.2	Reliability	8
5.2.1	Packet Loss	8
5.2.2	Temporary High Lag	8
5.2.3	Disconnects	8
5.2.4	Error Handling	8
5.3	Performance	8
5.3.1	Response Time	8
5.3.2	CPU Usage	8
5.3.3	Connected Players	8
5.3.4	Worst Case Scenario	9
5.3.5	Lag / Latency	9
5.4	Maintainability	9
5.4.1	Code Convention	9
5.4.2	Simple Maintainability and Reusable Code	9
5.4.3	Replaceable Low Level Network Framework	9
5.5	Documentation	9
5.6	Design Constraints	9
5.6.1	Limited Resources	9
5.6.2	Internet environment	9
6.	Appendix A – Glossary	10

Network Integration	Version: 0.95D
Software Requirements Specification	Date: 04/02/08
X4-NET-SRS	

## 1. Introduction

### 1.1 Purpose

The purpose of this SRS is to define the requirements for X4's network engine. It also provides an overview of the users which have to be taken into account as well as some information which cover the complete multiplayer integration rather than solely concentrating on the network engine in hope of providing the necessary background to identify and describe the requirements for the network engine.

### 1.2 Scope

The document covers only the requirements for the initial integration of the network engine into X4 (even though this initial integration should be considered feature complete). It will neither provide a requirement analysis of the complete multiplayer mode nor cover aspects which don't impact the design of the network engine.

### 1.3 Definitions, Acronyms, and Abbreviations

See Appendix A – Glossary and the “X4 Networking Background Information” document for an explanation of used definitions, acronyms and abbreviations.

### 1.4 References

Title	Report Number	Date	Publisher
X4 – Networking Background Information	Revision 1.4D	04/02/2008	Egosoft
doc\diplomathesis\X4 - Network Integration - Background 1.4D.pdf			

### 1.5 Overview

Chapter 2 covers the vision which stands behind the idea of a network engine followed by the Stakeholder requests for the engine. Chapter 3 provides the background knowledge necessary to understand the requirements as they are presented here. To provide some insight into the planned multiplayer mode, the use case model in chapter 4 provides a quick overview of expected user actions. Chapter 5 finally defines the requirements.

Network Integration	Version: 0.95D
Software Requirements Specification	Date: 04/02/08
X4-NET-SRS	

## **2. Vision**

### **2.1 Introduction**

All previous X – based games were pure single player games. They all lacked any kind of multiplayer capability.

Therefore it's not really surprising that one of the most frequently requested features by fans for the next sequel of the game is the ability to play against/with other people over the Internet. Some of the players were so desperate that they even thought about the idea of writing an X3 mod themselves with the intention of adding at least some basic multiplayer features.

Pure single player games all suffer from the same limitation: It's fun to play them a couple of times (and in case of the X based games even much longer than most other ones, because of its dynamic universe which changes all the time even without any influence of multiple players) but in the end they all lack the interaction with other people.

It's not only about being able to communicate in-game with your friends or directly be able to play and/or compete with them, but it's also a fact that no matter how intelligent your AI is, no NPC could cope with the dynamic a real player induces into the game environment.

You can see lots of examples where games kept their popularity for several years, simply because of their multiplayer mode. Just take Warcraft 3, Half-Life 2 or Diablo 2 for example. All of these games provide a single player mode (and can even be fun to be played for a long time without having an Internet connection at all). However, their huge popularity is mainly based on their integrated feature to compete with other players.

### **2.2 Positioning**

Providing multiplayer is what we also plan to achieve with X4 for the X-based games. Players being able to enjoy sharing their accomplishments with others, spending time together with their friends in-game and even founding clans to get together with people around the globe will increase the overall cost-benefit-ratio of X4.

### **2.3 Project goal and scope**

The project's goal is to provide the fundament, needed to be able to incorporate any kind of multiplayer capability into the current X4 code. That also means that implementing any concrete multiplayer game mode and UI or game design feature used to establish the connection between players is outside the scope of this project. Solely the required network features to be able to establish a connection will be provided.

Though the initial network engine won't be tested for cross platform operability either, the engine itself should not be restricted to allow players of the same platform being connected to one another, only.

Furthermore any kind of persistent features (like items on a per account basis or character development) is outside the scope of the network engine implementation, as are any in-game multiplayer UI elements (for instance chat windows, transaction windows, etc.).

Network Integration	Version: 0.95D
Software Requirements Specification	Date: 04/02/08
X4-NET-SRS	

## 2.4 User Descriptions

The following list provides an overview of all those who will work more or less directly with the network engine.

Name	Description	Responsibilities
Network Engine Developers	Develop, extend and maintain the Network Engine.	Implements the system.
Game Engine Developers (None-Network-Related)	Use functionality provided by the system and adjust their code according to the requirements of the Network Engine.	Defines required interaction points between the network and the game engine.
Mission Designers	Use the Network Engine to design multiplayer capable missions.	Specifies necessary features related to mission design.
Players	Indirect users of the engine since players use the multiplayer mode which relies on the network engine.	Late evaluation of the multiplayer mode (during Alpha- and Beta-Testing).

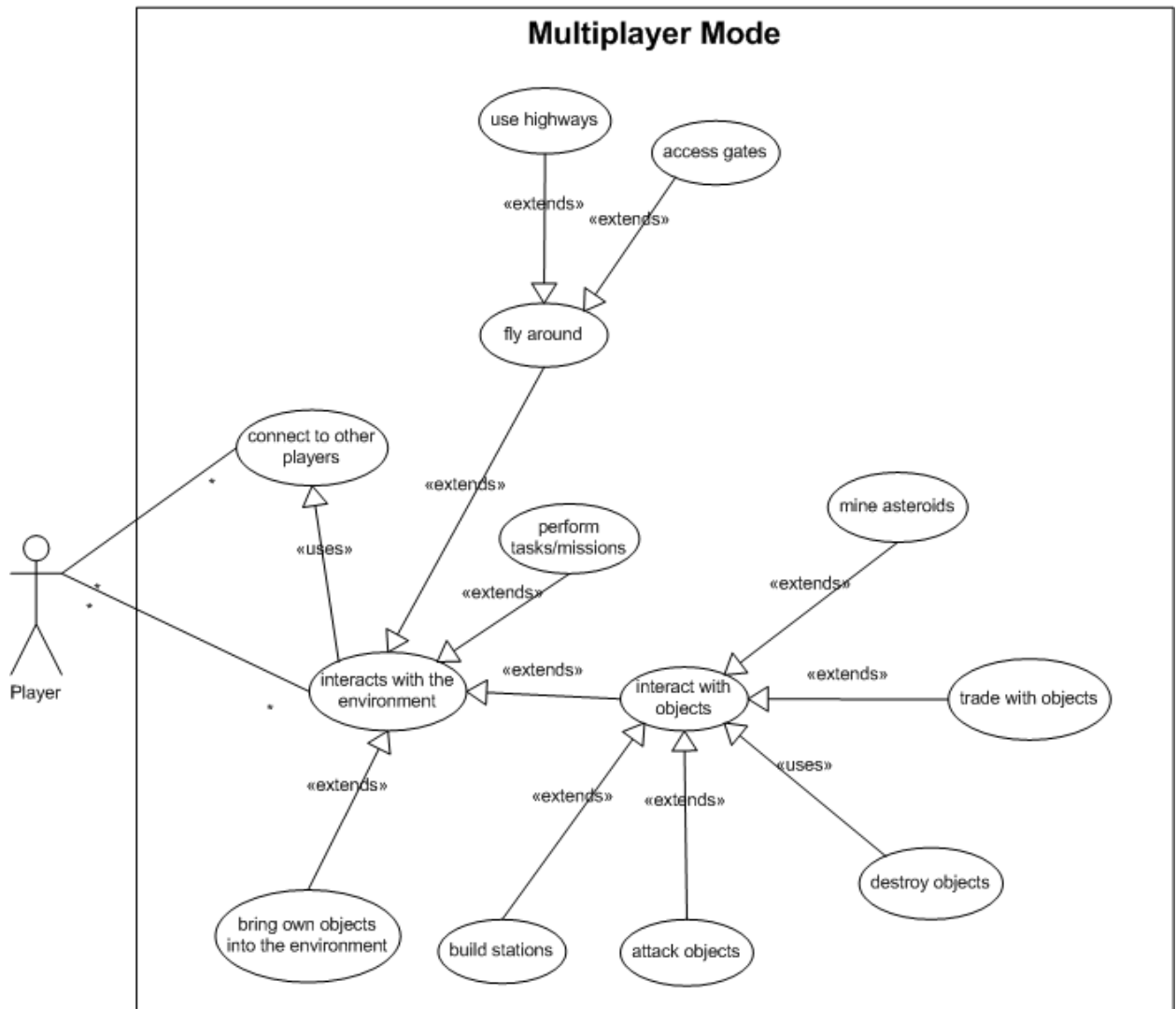
## 3. Overall Description

Refer to the X4 - Networking Background Information document for a complete description of the underlying backgrounds.

Network Integration	Version: 0.95D
Software Requirements Specification	Date: 04/02/08
X4-NET-SRS	

#### 4. Use-Case Model

It was decided that a network engine use case model is not applicable for clarifying the requirements. Instead a use case model describing the capabilities a player will have in a multiplayer game has been included and will be used to ensure that the architecture of the network engine covers all necessary features.



Network Integration	Version: 0.95D
Software Requirements Specification	Date: 04/02/08
X4-NET-SRS	

## **5. Requirements**

### **5.1 Usability**

#### **5.1.1 *Game Engine Developers***

Game developers must not be required to adjust major parts of their code to meet any requirements specified by the network engine. Unavoidable code adjustments need to be well documented and easy to understand without any need of detailed network engine knowledge.

#### **5.1.2 *Mission Designers***

Mission designers should be given all necessary tools (via the Mission Director) to provide them to write multiplayer missions. This functionality should be provided transparently, so the mission designers should not need to know anything network engine related at all.

### **5.2 Reliability**

#### **5.2.1 *Packet Loss***

The network engine should be capable to handle minor packet loss (up to 2%). It should still work even if the packet loss occurs over a longer period of time and to its best abilities allow players to still be able to play the game.

#### **5.2.2 *Temporary High Lag***

Temporary lag spikes should be coped with without causing any disconnects.

#### **5.2.3 *Disconnects***

Any sudden disconnect should be handled without interrupting the game play of the other users. The disconnected player should be able to reconnect to the server at least within a specific time frame and continue playing the game from the point on when he got disconnected.

#### **5.2.4 *Error Handling***

The network engine must be able to handle errors internally. It may however call the game engine's fatal error method at last resort.

### **5.3 Performance**

#### **5.3.1 *Response Time***

Response time of the network engine must not be limited by the game engine in any way. In fact it's desired to be able to send/receive updated packages up to 60 times / second.

#### **5.3.2 *CPU Usage***

The network engine must not hold up the complete game, although it is acceptable if the network engine causes a slight slow down of other parts of the game engine to sustain reliable data transmissions.

#### **5.3.3 *Connected Players***

The engine should be able to handle up to 8 players within a universe (but should not prevent this limit to be increased).



Network Integration	Version: 0.95D
Software Requirements Specification	Date: 04/02/08
X4-NET-SRS	

#### 5.3.4 *Worst Case Scenario*

The network engine should still be able to work under the following worst case scenarios:

Scenario 1: 8 connected players are flying inside the same crowded city while being in dogfight.

Scenario 2: 8 connected players are flying inside a different city, each.

#### 5.3.5 *Lag / Latency*

The experienced lag/latency must be kept on a low profile. The engine should be able to handle up to 8 players interacting in the same universe. 200 ms lag and a maximal upstream of about 10 kb/s should not hinder them to experience a fluent game play.

### 5.4 **Maintainability**

#### 5.4.1 *Code Convention*

Our code convention must be strictly followed.

#### 5.4.2 *Simple Maintainability and Reusable Code*

The engine must be easily maintainable in the future and minimize the required time to make necessary adjustments as well as keep the possibility to reuse it in any future game.

#### 5.4.3 *Replaceable Low Level Network Framework*

Any framework which handles the low level part of the network must be easily replaceable.

### 5.5 **Documentation**

The engine will be documented using a DoxyGen generated manual. Additional documentation which describes adjustments to other parts of the game engine needed for a complete integration of the network functionality must be provided (only applicable if any).

### 5.6 **Design Constraints**

#### 5.6.1 *Limited Resources*

We won't be able to increase the number of developers working on network related code, since all developers have already a strict timeframe to work with.

#### 5.6.2 *Internet environment*

We want the game to be playable over the internet. Therefore the transmission between two computers needs to be based on the Internet Protocol.

Network Integration	Version: 0.95D
Software Requirements Specification	Date: 04/02/08
X4-NET-SRS	

## 6. **Appendix A – Glossary**

NPC – Non-Player-Character