

Shattered Seal: Echelon

Requirements Document

Bryan Evans, bevans@my.fit.edu

Dan Baumann, dbaumann@my.fit.edu

Paul Graham, pgraham@my.fit.edu

Scott Leierer, sleierer@my.fit.edu

Stephen Garcia, sgarciam@my.fit.edu

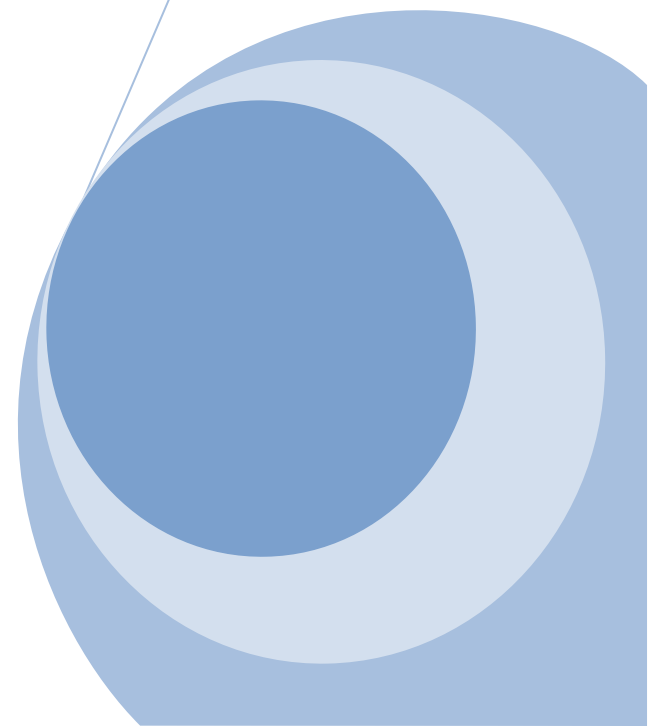


Table of Contents

Table of Contents	i
Revision History	ii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Intended Audience and Reading Suggestions	1
2. Overall Description.....	1
2.1 Product Perspective.....	1
2.2 Product Features.....	1
2.3 User Classes and Characteristics.....	1
2.4 Operating Environment.....	1
2.5 Design and Implementation Constraints	2
2.6 User Documentation	2
3. Game Mechanics Requirements	2
3.1 Artificial Intelligence	2
3.2 Models.....	2
3.3 Other Game Mechanics.....	3
4. Gameplay Requirements	4
4.1 User Options	4
4.2 User Interactions with the Game Environment	4
4.3 Mission Objectives.....	5
4.4 Other Gameplay Requirements.....	6
5. Interface Requirements	7
5.1 Graphical User Interfaces.....	7
5.1.1 Main Menu.....	7
5.1.2 Pregame Interface	7
5.1.3 In-Game Interface	8
5.1.4 Postgame Interface.....	9
5.2 Hardware Interfaces	10
5.3 Software Interfaces	11
6. Audio Requirements	11
7. Other Nonfunctional Requirements.....	11
7.1 Performance Requirements	11
7.2 Software Quality Attributes	12
8. Other Requirements	12
Appendix A: Glossary.....	13

1. Introduction

1.1 Purpose

This requirements document is being written for Shattered Seal: Echelon, an Xbox 360 title being developed for a course at the Florida Institute of Technology.

1.2 Intended Audience and Reading Suggestions

This document is intended primarily for the use by the developers while planning the design and implementation of the software. Secondly, this document will be used by professors of the Florida Tech Department of Computer Sciences to assess the development team's progress. Assessments may also include those from other Florida Tech students.

2. Overall Description

2.1 Product Perspective

Shattered Seal: Echelon is a new product which uses the Xbox 360 architecture and the XNA libraries to make it compatible with the Xbox 360 console. This product requires an Xbox 360 and an Xbox 360 controller to operate entirely as expected, however it may be possible to run this product on a PC with similar graphical and processing capabilities as well as a connected Xbox 360 controller.

2.2 Product Features

The main feature of our software is that it is a fully interactive game. The user must progress through a series of levels in order to complete the game. The game itself is a top-down shooter, and as such, has the majority of gameplay as having the user fending off swarms of enemy units.

2.3 User Classes and Characteristics

The primary user class that will use this product is the Xbox 360 gamer. These users have a wide variety of frequency of use but besides that they will all have relatively the same experience while using the product. It is impossible to know for sure who exactly will be using the product so it is important to make the product easy to use for a user of any level of education, experience, and expertise.

2.4 Operating Environment

The software is being designed to be run on the Xbox 360 System Software. This is the upgradeable operating system that is run on the Xbox 360 platform. The version of this operating system that we are designing for is 2.0.8498.0.

2.5 Design and Implementation Constraints

Design is limited to the Xbox 360 console and the Windows operating systems. Development itself must use XNA, and the final product will be released through the Creator's Club as an available game on the Xbox Live Marketplace.

As part a project for a course at Florida Tech, the design team is limited to two semesters (roughly 8-9 months) to finish the product.

2.6 User Documentation

As this product will only be distributed in digital, the documentation and tutorials will be limited to in-product help. There will be a tutorial available for new users. Also, a diagram of the control scheme may be viewed from the menu screen to help users familiarize themselves with controls. Any other assistance that users may require will be given during use of the product.

3. Game Mechanics Requirements

The game mechanics are what go on behind the scenes. They function as the rules and building blocks that the gameplay then builds off of. The majority of game mechanics for this software will consist of AI, in-game physics, and the modeling.

3.1 Artificial Intelligence

Requirement	Name	Release History
3.1.0.0	Enemy AI	1.0 (September 18, 2009)
Description: Enemy characters must implement algorithms that imitate human decisions. Although basic, the algorithms must vary for the different types of enemy units in order to add a level of difference in the gameplay of combating the various enemies.		
Priority: 5	Dependencies: none	Type: functional
3.1.0.1	Anti-trap Measures	1.0 (September 18, 2009)
Description: The AI must include measures to prevent enemy units from getting trapped in certain locations while trying to pursue the player character. Such situations include getting stuck behind structures like walls and buildings.		
Priority: 5	Dependencies: 3.1.0.0	Type: non-functional

3.2 Models

Requirement	Name	Release History
3.2.0.0	2D Models	1.0 (September 18, 2009)

Description: Graphically, the game will be 2D with representation through sprites versus being 3D. As such, the models will only interact in the x and y-axis.		
Priority: 5	Dependencies: <i>none</i>	Type: non-functional
3.2.0.1	Lighting Effects	1.0 (<i>September 18, 2009</i>)
Description: Because the virtual world of Shattered Seal will be represented with 3D models, realistic, albeit basic, physics must be applied to the lighting effects on the models themselves. Thus, objects will cast shadows if only lit from one side.		
Priority: 2	Dependencies: 3.2.0.0	Type: functional
3.2.0.2	Enemy and User Models	1.0 (<i>September 18, 2009</i>)
Description: There must be different models for the player character and the enemy characters that he/she will encounter. Enemy characters may use the same model, but they must be unique between the different types of enemy.		
Priority: 4	Dependencies: 3.2.0.0	Type: non-functional
3.2.0.3	Scenery Models	1.0 (<i>September 18, 2009</i>)
Description: The background scenery must be represented in 3D. Certain objects on the map must be models. Also, the scenery must be decorated with an adequate level of detail that a user will know what the models are supposed to represent.		
Priority: 2	Dependencies: 3.2.0.0	Type: non-functional

3.3 Other Game Mechanics

Requirement	Name	Release History
3.3.0.1	Basic Combat Physics	1.0 (<i>September 18, 2009</i>)
Description: The game must implement a basic level of physics, specifically during combat. This specifically refers to the movements and qualities exhibited by casted spells and fired weapons. This includes, but is not limited to, gravity, bouncing off of objects, and inertia.		
Priority: 5	Dependencies: <i>none</i>	Type: functional
3.3.0.2	Kill Statistics	1.0 (<i>September 18, 2009</i>)
Description: The game must keep a career total for the number of kills the user has achieved for the save file.		
Priority: 2	Dependencies: <i>none</i>	Type: functional

4. Gameplay Requirements

The gameplay requirements are more specifically related to the user's interactions with the game other than what happens through interfaces. They dictate how the user will have to play the game. Gameplay itself is what helps make games unique from each other.

4.1 User Options

The user options are the features of gameplay that offer customizability in how the user approaches and plays the game.

Requirement	Name	Release History
4.1.0.1	Multiple Usable Weapons	1.0 (September 18, 2009)
Description: The gameplay features a variety of weapons that the user has the option to equip. The weapons vary in their behavior. This includes the how and what they fire.		
Priority: 3	Dependencies: none	Type: non-functional
4.1.0.2	Multiple Usable Spells	1.0 (September 18, 2009)
Description: The gameplay features a variety of spells that the user has the option to utilize. These spells are divided into two categories: offensive and defensive spells.		
Priority: 5	Dependencies: none	Type: non-functional
4.1.0.3	Multiple Usable Tech Abilities	1.0 (September 18, 2009)
Description: The user will have a variety of different "tech" abilities that alter the physical laws around him. These abilities will be reusable throughout the mission, but will have a cool down between uses.		
Priority: 4	Dependencies: none	Type: non-functional
4.1.0.4	Talent Trees	1.0 (September 18, 2009)
Description: The user will have seven talent trees where they will be able to spend points to improve abilities, weapons, spells, and to acquire new ones.		
Priority: 4	Dependencies: none	Type: non-functional

4.2 User Interactions with the Game Environment

The following features and requirements are related to in-game interactions within the game world that surrounds the player character.

Requirement	Name	Release History
4.2.0.1	Movable Player character	1.0 (September 20, 2009)

Description: The player character should be able to move around in the game environment. To represent this through the In-Game UI, the player character will remain in the center of the screen while the world around it moves.		
Priority: 3	Dependencies: <i>none</i>	Type: functional
4.2.0.2	Anti-Grinding Measures	1.0 (September 19, 2009)
Description: Grinding is a term used to describe repetitive gameplay in order for the user to gain experience. The gameplay must be designed to avoid this approach by not using endless waves of enemies.		
Priority: 3	Dependencies: <i>none</i>	Type: non-functional
4.2.0.3	Hurt Enemies	1.0 (September 19, 2009)
Description: Enemy units will possess a health attribute that can be lowered by actions the user makes.		
Priority: 5	Dependencies: <i>none</i>	Type: functional
4.2.0.4	Move Through Enemy Dead Units	1.0 (September 19, 2009)
Description: When enemy units run out of health, the player character must be able to pass through the object in order to prevent the user from being trapped.		
Priority: 5	Dependencies: <i>none</i>	Type: functional
4.2.0.5	Boss Battles	1.0 (September 20, 2009)
Description: A boss battle will occur near the end of each mission. These battles will limit the field of play for the user, and they will also rely more on scripted moves rather than enemy AI.		
Priority: 3	Dependencies: <i>none</i>	Type: non-functional

4.3 Mission Objectives

Mission objectives provide the user with a set of goals during gameplay. These goals are driven by the storyline of the game. As such, mission objectives are an important aspect of enveloping the user in the game experience.

Requirement	Name	Release History
4.4.0.0	Mission Objectives	1.0 (September 20, 2009)
Description: Missions must be designed with specific objectives that must be completed before the moving on to the next mission.		
Priority: 3	Dependencies: <i>none</i>	Type: non-functional
4.4.0.1	Optional Objectives	1.0 (September 20, 2009)
Description: Missions should also include objectives that are optional and therefore not required to complete before moving onto the next mission. If completed, these objectives will grant the user with bonuses and other rewards.		

Priority: 3	Dependencies: 4.4.0.0	Type: non-functional
4.4.0.2	Optional Objective Replay	1.0 (September 20, 2009)
Description: Upon finishing the game, users can replay missions and complete uncompleted, optional mission objectives.		
Priority: 1	Dependencies: 4.4.0.1	Type: non-functional
4.4.0.3	Objective Completion Notification	1.0 (September 20, 2009)
Description: Upon completing an objective during gameplay, there must be a notification to the user of the event. This notification will consist of both an audible and a visual signal. Also, notifications are given for both optional and mandatory objectives.		
Priority: 3	Dependencies: 4.4.0.0	Type: functional

4.4 Other Gameplay Requirements

Requirement	Name	Release History
4.4.0.1	No Quick Time Events	1.0 (September 20, 2009)
Description: Quick time events are a type of gameplay wherein the user must time button pressing according to onscreen prompts. The gameplay must not implement this method of gameplay.		
Priority: 1	Dependencies: <i>none</i>	Type: non-functional
4.4.0.2	Changing Mana	1.0 (September 20, 2009)
Description: There must be an attribute that represents how much mana is remaining. This attribute will change (specifically, it will decrease) when spells are casted.		
Priority: 5	Dependencies: <i>none</i>	Type: functional
4.4.0.3	Changing Health	1.0 (September 20, 2009)
Description: There must be an attribute that represents how much health is remaining. This attribute will change as the player character injured or healed.		
Priority: 5	Dependencies: <i>none</i>	Type: functional
4.4.0.4	Multiple Save Files	1.0 (September 20, 2009)
Description: A save file is a copy of the current state of the game. The game must allow the user to save multiple save files if he/she chooses to do so.		
Priority: 5	Dependencies: <i>none</i>	Type: functional

5. Interface Requirements

5.1 Graphical User Interfaces

The graphical user interfaces (GUI) are the means the user use to will interact with the game. As such, they are broken down into the following categories: main menu, pregame, in-game, postgame.

5.1.1 Main Menu

The Main Menu is the user interface that the user will use when the game is initialized.

Requirement	Name	Release History
5.1.1.0	Main Menu	1.0 (September 16, 2009)
Description: The game must have a main menu to allow the user to navigate prior to beginning any gameplay.		
Priority: 5	Dependencies: none	Type: non-functional
5.1.1.1	Menu with Logo	1.0 (September 16, 2009)
Description: The main menu should feature the team logo somewhere.		
Priority: 1	Dependencies: 5.1.1.0	Type: non-functional
5.1.1.2	Load Feature on Menu	1.0 (September 16, 2009)
Description: From the main menu, the user should be able to load a save file if available.		
Priority: 3	Dependencies: 5.1.1.0	Type: functional
5.1.1.3	Exit Game	1.0 (September 18, 2009)
Description: The user must be given the option to exit the game and return to the Xbox 360 dashboard.		
Priority: 5	Dependencies: 5.1.1.0	Type: functional

5.1.2 Pregame Interface

The Pregame Interface is what the user must navigate prior to beginning the actual gameplay.

Requirement	Name	Release History
5.1.2.1	Mission Briefing	1.0 (September 18, 2009)
Description: The pregame interface must include a mission briefing for each level. Each mission briefing must be unique. Also, they will be told in story format as they serve to provide information about the occurrences in the world of Shattered Seal. Mission briefings must also inform the user of the mission objectives available for the level at hand.		
Priority: 3	Dependencies: 5.1.2 4.4.0.0	Type: non-functional

5.1.2.2	Mission Objective Screen	1.0 (September 18, 2009)
Description: After the mission briefing, there must be a screen that lists the mission objectives that were mentioned during the briefing.		
Priority: 3	Dependencies: 5.1.2 4.4.0.0	Type: non-functional
5.1.2.3	Choose Loadout	1.0 (September 18, 2009)
Description: The interface must provide the user with the ability to choose their loadout for the upcoming level. The loadout is what weapon from the user's arsenal he/she wishes to arm their player character before beginning gameplay.		
Priority: 3	Dependencies: 5.1.2 4.1.0.1 4.1.0.2	Type: functional

5.1.3 In-Game Interface

The In-Game Interface is essentially what the user will interact with most of the time. As such, it is important for this interface to not overwhelm the user while still providing enough information and options for he/she to be satisfied.

Requirement	Name	Release History
5.1.3.0	In-Game UI	1.0 (September 16, 2009)
Description: The game must have provide a fully functional user interface for the user to operate the game during the actual gameplay.		
Priority: 5	Dependencies: none	Type: non-functional
5.1.3.1	Health Bar	1.0 (September 16, 2009)
Description: The interface should feature a colored bar that represents the player character's health. This bar should indicate the amount of health remaining and function in real-time.		
Priority: 4	Dependencies: 5.1.3.0	Type: functional
5.1.3.2	Mana Bar	1.0 (September 16, 2009)
Description: The interface should feature a colored bar that represents the player character's mana. This bar should indicate the amount of mana remaining and function in real-time.		
Priority: 4	Dependencies: 5.1.3.0	Type: functional
5.1.3.3	Character Portrait	1.0 (September 16, 2009)
Description: The interface should feature a portrait image of the player character.		
Priority: 2	Dependencies: 5.1.3.0	Type: non-functional
5.1.3.4	Display Ability Cooldown	1.0 (September 16, 2009)
Description: When the user uses a tech ability, there is a cooldown		

period. The interface should feature some sort of countdown feature that shows how much time the cooldown has remaining. Also, this display should be over the player character's portrait.		
Priority: 2	Dependencies: 5.1.3.0 5.1.3.3	Type: functional
5.1.3.5	Control Scheme Display	1.0 (September 21, 2009)
Description: When the user holds the Back button, a window will pop up and display a diagram showing the control scheme of the game via the Xbox 360 controller.		
Priority: 2	Dependencies: none	Type: functional

5.1.4 Postgame Interface

The Postgame Interface appears upon the completion or failure of a mission.

Requirement	Name	Release History
5.1.4.1	Mission statistics	1.0 (September 18, 2009)
Description: The postgame interface should display statistics accrued by the user during his/her actions during the mission. This should include kills, completion time, and completed mission objectives.		
Priority: 3	Dependencies: 5.1.4 3.3.0.2	Type: functional
5.1.4.2	Option to Quit	1.0 (September 18, 2009)
Description: The postgame interface must provide the user with the option to quit the game. Selecting the quit option should take the user to the Main Menu interface.		
Priority: 5	Dependencies: 5.1.4	Type: functional
5.1.4.3	Option to Retry	1.0 (September 18, 2009)
Description: The user is given the option to restart the mission. Kill statistics gained through previous attempts at the mission will still contribute to the overall statistics of the user's career totals. However, the statistics from previous attempts of the mission will not contribute to the overall mission statistics once the mission is completed. If the retry option is selecting after failing the level, any objectives completed during the attempt at the mission will be reset. Otherwise, completed objectives will be saved.		
Priority: 4	Dependencies: 5.1.4	Type: functional
5.1.4.4	Ability to Save Game	1.0 (September 18, 2009)
Description: In-between missions, the user should be able to save his/her game status. This will save the invested points in the player character's talent, spell, and weapon trees. The save will also preserve the user's career statistics as well as allow he/she to continue playing where he or she saved.		
Priority: 4	Dependencies: 5.1.4	Type: functional

5.1.4.5	Level Up Interface	1.0 (September 18, 2009)
Description: Following the completion of a mission, the user should be given the opportunity to invest earned points into talents, spells, and weapons. The new level should also be indicated.		
Priority: 5	Dependencies: 5.1.4	Type: non-functional

5.2 Hardware Interfaces

Because the game will be run on the Xbox 360 platform, the software will have an interface between our product and the Xbox 360 controller. The XNA toolset will provide the adequate commands to successfully communicate between the two if planned correctly.

Requirement	Name	Release History
5.2.0.1	Controller Scheme	1.0 (September 16, 2009)
Description: There must be a working control scheme for the Xbox 360 controller. The control scheme must implement controls that are ergonomic as according to human factors engineering.		
Priority: 5	Dependencies: none	Type: non-functional
5.2.0.2	Left Analog Stick for Movement	1.0 (September 17, 2009)
Description: The left analog stick of the Xbox 360 controller will be the input device for the user to control the movements of the player character. In by keeping with common practice in the industry, pushing forward on the stick will move the character towards the top of the screen. Likewise, pushing left, down, and right will move the character towards the left, bottom, and right of the screen, respectively. Also, the inputs are not limited to up, down, left, and right, but rather, include all 360 degrees.		
Priority: 5	Dependencies: 5.2.0.1	Type: functional
5.2.0.3	Right Analog Stick for Aiming and Shooting	1.0 (September 17, 2009)
Description: The right analog stick of the Xbox 360 controller will be the input device for the user to control the aiming and shooting of the player character (note: shooting includes both weapons and spells). In by keeping with common practice in the industry, pushing forward on the stick will both aim and fire towards the top of the screen. Likewise, pushing left, down, and right will both aim and shoot towards the left, bottom, and right of the screen, respectively. Also, the inputs are not limited to up, down, left, and right, but rather, include all 360 degrees.		
Priority: 5	Dependencies: 5.2.0.1	Type: functional

5.3 Software Interfaces

Software interfaces consist of the connection between *Shattered Seal: Echelon* and other software components, such as those found on the Xbox 360. Specifically, this includes tools and libraries.

Requirement	Name	Release History
5.3.0.1	Use XNA	1.0 (September 16, 2009)
Description: The software must be written using Microsoft XNA to communicate with the Xbox 360.		
Priority: 5	Dependencies: <i>none</i>	Type: constraint

6. Audio Requirements

Requirement	Name	Release History
6.0.0.1	Sound Effects	1.0 (September 20, 2009)
Description: There must be appropriate sound effects for the in-game objects. Sound effects will be used when in-game objects interact with each other, such as two characters attacking each other.		
Priority: 3	Dependencies: <i>none</i>	Type: functional
6.0.0.2	Background Music	1.0 (September 20, 2009)
Description: Background music must change with the storyline. For example, boss battles will use different background music than what is played while running through the majority of the missions.		
Priority: 3	Dependencies: <i>none</i>	Type: functional

7. Other Nonfunctional Requirements

7.1 Performance Requirements

Requirement	Name	Release History
7.1.0.1	100 Millisecond Response Time	1.0 (September 16, 2009)
Description: Response time is an extremely important factor to this type of software. As such, limiting the tolerable amount of latency is very important to having software that will satisfy the user.		
Priority: 3	Dependencies: <i>none</i>	Type: non-functional

7.2 Software Quality Attributes

Requirement	Name	Release History
7.2.0.1	High Reliability	1.0 (<i>September 16, 2009</i>)
Description: The software must maintain a high level of reliability in not crashing.		
Priority: 5	Dependencies: <i>none</i>	Type: non-functional

8. Other Requirements

Requirement	Name	Release History
8.0.0.1	Use C#	1.0 (<i>September 18, 2009</i>)
Description: The software must be programmed in C#.		
Priority: 3	Dependencies: <i>none</i>	Type: constraint
8.0.0.2	Playable on the Xbox 360	1.0 (<i>September 18, 2009</i>)
Description: The software is being specifically designed to run on the Xbox 360. Thus, the final product must be able to run on this platform.		
Priority: 5	Dependencies: <i>none</i>	Type: constraint

Appendix A: Glossary

A

AI - Artificial Intelligence. This is the programmed attempt to simulate human decision making processes. In video games, however, AI refers to the illusion of intelligence based on the behaviors of non-player characters.

Analog Stick - a type of input device found on most modern game controllers (typically, these controllers feature two and are referred to as dual analog sticks). Analog sticks provide two-dimensional input, but they can also be pressed downwards as a button. Analog sticks are a form of joystick.

B

Boss Battle - an enemy-based challenge located at the climax of a mission. Boss Battles are typically scripted fights in that the boss's movements and actions follow a pattern.

C

C# - a .NET compliant language developed by Microsoft. C# is a combination of C++ and Java, amongst other languages.

Cooldown - the period of time that a user must wait before using the same ability again.

G

Grinding - a term in video games that describes a style of gameplay wherein the player must perform repetitive, dull actions in order to achieve success and/or access certain features.

L

Loadout - the equipment, weapons, etc., selected for use by the character.

M

Mana - an adopted term in the world of fantasy for magic points. These magic points are required for casting spells.

P

Player character - the character on the screen that the user controls. In Shattered Seal: Echelon, the player character is always located in the very center of the screen.

Q

Quick Time Event - a type of gameplay wherein the user must time button pressing according to onscreen prompts.

S

Save File - also known as a 'game save', it is a set of data regarding the user's progress in the game's storyline.

Sprite - a two-dimensional image or animation used to represent an in-game object, character, etc. Sprites are often a collection of different images that give the illusion of animation and sometimes even a third dimension based on how these images are displayed (such as the order and timing).

T

Top-down Shooter - also known as "Shoot 'em Up," this is a subgenre of shooter video games. In a top-down shooter, the user controls a single character as they get attack by waves of enemies. The perspective of the gameplay is from high above the player character and thus provides the user with a full field of vision around his/her character.

X

Xbox 360 - a video game console produced by Microsoft. This is one of three consoles that represent the seventh generation of consoles.

XML - Extensible Markup Language. A set of rules for encoding documents electronically.

XNA - Xbox New Architecture. Also known as Microsoft XNA, this is a set of tools provided by Microsoft for game developers working on Xbox 360 games. The libraries are primarily intended to be used with C#.