



**World Exposition Shanghai China 2010**

**Expo Shanghai Online**

**SUGGESTIONS FOR DEVELOPMENT AND  
CONSTRUCTION OF EXPERIENCING PAVILION**

**1<sup>st</sup> Edition**

**Bureau of Shanghai World Expo Coordination**

**August 2009**



# Preface

The *World Exposition Shanghai China 2010 Online -- Suggestions for Development and Construction of the Experiencing Pavilion* (hereinafter referred to as the *Suggestions*) is formulated to guide and assist participants in the development and construction of the Experiencing Pavilion of World Exposition Shanghai China 2010 Online (hereinafter referred to as “Experiencing Pavilion”).

Made in Chinese and English versions, this document features measures proposed by the Organizer to optimize the Experiencing Pavilion performance and to deal with other technical aspects in association with an Experiencing Pavilion. Along with further progress of the Expo Online Project, the Organizer will update this document on a timely and continuous basis to ensure that all necessary information will be promptly provided for Participants.

Should you have any question or comment regarding the Suggestions, please feel free to contact us:

**Contacts:** Mr. Chen Xuyan Tel: 86-21-22062079

Mr. Shen Feifei Tel: 86-21-22062264

**Address:** No. 3588, Pudong South Rd, Shanghai, China

Expo Shanghai Online Program Office

Communication and Promotion Department

Bureau of Shanghai World Expo Coordination

**Postal code:** 200125

**E-mail:** tech@expo2010.gov.cn

**Fax:** 86-21-22060311

**Official website:** <http://www.expo2010.cn>

**Website for Expo Shanghai Online:** <http://www.expo.cn>

**Website for Collaborative Working Platform:** <http://pm.expo.cn>



# Table of Contents

<b>1</b>	<b>OVERVIEW.....</b>	<b>1</b>
<b>2</b>	<b>OPTIMIZATION OF EXPERIENCING PAVILION PERFORMANCE .....</b>	<b>2</b>
2.1	VERSION DESIGN.....	2
2.1.1	<i>Multi-versions of Experiencing Pavilion .....</i>	<i>2</i>
2.1.2	<i>Standalone/offline Version of Experiencing Pavilion.....</i>	<i>2</i>
2.2	EXPERIENCING PAVILION DEPLOYMENT .....	2
2.3	EXPERIENCING PAVILION OPTIMIZATION.....	3
2.3.1	<i>Development with Flash .....</i>	<i>3</i>
2.3.2	<i>Development with 3DVIA Virtools.....</i>	<i>5</i>
<b>3</b>	<b>OTHERS.....</b>	<b>11</b>
3.1	DATA LOADING PROMPT PAGE .....	11
3.2	DESIGN OF INTERACTIONS.....	11



# 1 Overview

A very new project in the history of World Expo, Expo Online initiative is in the face of many challenges, including the unbalanced global network distribution, differences in end users' network conditions, difficulty in accurate estimations of concurrent visits, and high bandwidth requirements to support 3D presentations, all of which may cause bad user experience.

In light of the case, the "Suggestions" is formulated by the Organizer based on the "*Guide for Development and Construction of the Experiencing Pavilion*". The difference is that participants must follow the requirements stated in the "Guide", while referring to the "Suggestions" based on real situations to improve efficiency and quality, and ensure fast and easy access.

The "Guide" focuses on optimization of the Experiencing Pavilion performance and other related aspects, proposing solutions to differences in end users' network conditions and difficulties in accurate estimation of concurrent visits, and solutions in interactive function designs to address the restriction of data support for the Experiencing Pavilion.

## 2 Optimization of Experiencing Pavilion Performance

### 2.1 Version Design

#### 2.1.1 Multi-versions of Experiencing Pavilion

According to the “Guide”, the Experiencing Pavilion should be accessed with a bandwidth of 256Kbps, and will be easily accessed if with 512Kbps. In the case of poor network conditions of end users or huge web traffic, some users may have a hard time accessing the Experiencing Pavilion.

When constructing the Experiencing Pavilion, participants can include two versions of the Experiencing Pavilion in the outcome file package: a standard-quality version and a low-quality one.

The standard version should be constructed in strict accordance with the “Guide” to ensure the access of users with favorable network capacity.

With the low-quality version, however, the number and size of files should be minimized to alleviate the network burden when downloading assets so as to ensure the easy access of users with poorer network conditions.

#### 2.1.2 Standalone/offline Version of Experiencing Pavilion

In order to deal with poorer network capacity and large web traffic, participants can provide the outcome package of the standalone (offline) version so that visitors can access an Experiencing Pavilion already downloaded on PC. The downloading address of the standalone version can be informed on the homepage of the online pavilion for the convenience of visitors.

The outcome package for the standalone/offline version will not be restricted by the parameters stated in the “Guide”, but must be able to operate independently.

### 2.2 Experiencing Pavilion Deployment

CDN network will be adopted to distribute the Experiencing Pavilions for easy local access. However, due to limited resources and network conditions, visitors in some areas may still have the problem of slow access.

To deal with this, participants can deploy the contents of Experiencing Pavilion on their own websites (own official Expo Website and/or other relevant websites). The visiting

address can be informed on the homepage of the online pavilion for the convenience of visitors.

## 2.3 Experiencing Pavilion Optimization

According to the “Guide”, participants can choose Flash and/or 3DVIA Virtools to develop the Experiencing Pavilion. Detailed solutions and guidance are therefore given below to help participants to improve the development efficiency and quality.

### 2.3.1 Development with Flash

#### **2.3.1.1 Network Load Optimization**

##### **Optimization of Client Program Files**

Large client program files should be disintegrated. Separate functional modules from the main program and load certain module when necessary to downsize the main program and make it more flexible.

##### **Load Necessary Files with Streaming**

URLStream in Flash provides low-level access to download URL. Data can be instantly used by ActionScript once downloaded. This is totally different from using URLLoader, which can not be used until the complete file is downloaded. URLStream also allows the end of streaming before completing the download.

With streaming, the speed of loading files can be greatly improved. Since Flash Player loading is single-route, if users wants to view another animation while the program is already loading one, the program will close the loading animation and load the one that the user wants to view, as such, the bandwidth can be properly utilized.

##### **Use Pre-loading Mechanism**

When the current webpage is already loaded, and the network is idle, the Client can select privileged web pages for pre-loading. This will help users spend less time waiting and balance the network load.

#### **2.3.1.2 Optimization of Server Load**

##### **Reduce HTTP Requests**

More HTTP requests lead to concurrent pressure to the server and longer responding time. Reducing HTTP requests by decreasing the elements of a webpage can speed up the web page loading.

#### **2.3.1.3 Optimization of CPU Load**

##### **Use Bitmap Cache in Operation**

Bitmap Cache allows you to appoint an edited static image (e.g. background image) or a button to cache as bitmap in operation. In doing so, the performance of playback is optimized. Caching an edited image to bitmap can prevent Flash Player from repeating image description, thus greatly improve the performance of playback.

### **Use Less Filters**

Type, quantity and quality of filters applied to the object will affect the playing of SWF files. The more filters applied to the object, the larger quantity is needed for Adobe Flash Player to correctly display the created visual effect. Therefore, filters should be applied in a limited number to a given object.

Each filter contains widgets, which can adjust the intensity and quality of the applied filters. Lowering configuration for slow-running computers can improve performance. If contents are created to be shown on computers at different performance levels or the capacity of users' computers is uncertain, set the quality level as "low" for optimum playback performance.

### **Use More Simple-Structured Vector Graphics**

The more complex the vector graphic is, the more server resources are needed. Therefore, use less complex vector graphics, without breaking fonts apart, and use less excessive filling color, and reduce the deformed area of the graphic in a possible way.

### **Control Frame Rate of Animation**

Frame rate set for FLA files should be considered when adding animation to application programs. Frame rate may affect SWF file and the performance of computers to play the file. If the frame rate is set too high, the server may go wrong, especially when many sources are used or ActionScript is used to create animations. The frame rate of submitted online pavilion should not exceed 30fps.

### **Use Loop Code Cautiously**

Iteration in Flash can run very fast in Flash Player, but looping depends greatly on the server. The more the iterating times in the loop, the more codes in each module run, and the more server sources are used. Unreasonable looping may lead to poor performance or stability.

#### **2.3.1.4 Memory Optimization**

##### **Avoid Embedding Video in SWF File**

Embedding videos in SWF file may cause file size being too large. Flash Player will consume a great deal of memory and may even collapse while downloading or trying to play large video-embedded SWF files.

### **Use More Simple-Structured Vector Graphics**

The more complex the vector graphic is, the more server resources are needed. Therefore, use less complex vector graphics, without breaking fonts apart, and use less excessive filling color, and reduce the deformed area of the graphic in a possible way.

### **Recycle Wastes Promptly**

Useless objects should be recycled to the garbage bin to release used memory.

### **Use Bitmap Cache Cautiously**

Do not overuse the surface (as the cached display object is enabled). Memory for each surface is more than that of regular display objects, which means surface can be enabled only when the displaying performance needs to be improved.

Memory for cached bitmap is much more than those of regular display objects. For example, if Sprite on the stage is 250 x 250 pixels, it needs 250KB memory to cache it. However, if it is a regular (non-cached) Sprite, it needs only 1KB memory.

Avoid magnifying the cached surface. If bitmap cache is overused, especially in magnifying contents, huge memory will be used.

## 2.3.2 Development with 3DVIA Virtools

### **2.3.2.1 Optimization of Files**

#### **Multimedia Files**

Picture resolution is advised not to be greater than 512\*512 because picture size is the major element of final file size of your compositions. You should use JPG as your compression format if you will load external image files. Video format must be WMV. We recommend to use URL Stream Video as your Live Video instead of using other live video sources (camera, Web Cam, etc)

#### **File Path**

We recommend you to use dynamic loading techniques to load your audio, video or large NMO files. Please use “Web Download” + “Object Load” building blocks to load your files for online contents (require Extended & Connection Server). You can simply use “Object Load” building block for offline applications (pre-downloaded package, on-site applications, etc.) .

We recommend you to use relative path if your files are in the same folder as main composition file or its subfolders, otherwise use absolute path. For instance, assume that there are “main.vmo” and “ resource ” folder in “ssize” folder. When you load “sub1.nmo” file which is located in the “resource” folder, then the path should be “\resource\sub1.nmo”.

Pay attention to “Relative to URL” option of “Web Download”. When you check this option, it will convert relative path to URL. For example, if you specify “\resource\sub1.nmo” and your main composition is located at “http://ssize.pavilion.expo.cn/pavilion\_no/ssize/main.vmo”, then it will download the file located at “http://ssize.pavilion.expo.cn/pavilion\_no/ssize/sub1.nmo”.

When loading files outside of the folder containing the main composition, e.g. the “main.vmo” is in “ssize” folder and loading “sub2.nmo” file in “\size\resource” folder. The Web Download reference path should be “http://\size.pavilion.expo.cn/pavilion\_no/\size/resource/sub2.nmo”

### **File Size**

We recommend you to maintain the following size specifications of your media files:

- VMO (main interface) file should be smaller than 2MBexternal
- NMO files to be loaded dynamically should be smaller than 10MB
- Videos included in VMO/NMO files should be smaller than 5MB
- Live streaming videos should be smaller than 30MB
- The package you send to EXPO should not exceed 2GB

### **2.3.2.2 Model Optimization**

The following methods are recommended for optimizing models to compress files and enhance performance.

#### **Remove Invisible Faces**

It is recommended to remove faces that will never be visible. For example, you can remove faces under objects if camera never goes bellow, faces on top of very tall objects if the camera never goes higher, faces behind a wall if we never go behind, and faces inside objects if we never look inside, etc.

#### **Grouping Objects**

If there are plenty of objects in a little zone of the scene, we can merge them into a single object. This will simplify the scene hierarchy in Virtools, and it will make it easier to manage the scene and to improve the performance of rendering engine to check and sort 3D objects in the scene.

#### **Cutting Big Objects**

On the other hand, you can separate big objects by zones in the scene. When Virtools cameras look an object, the entire object (mesh) will be passed to the rendering engine even if only a very little part is visible. Therefore, if your object is too large and

complicated, then the rendering engine has to process many unnecessary faces and vertices

### **Subdivision Level**

Particularly for round objects or objects with smoothly curved surfaces, please choose the good subdivision levels. For this kind of object, optimization allows you to reduce the number of faces, and hence reduce your data size and improve the performance significantly.

### **3D Sprites**

A 3D Sprites is a billboard (plane consists of 2 faces) that can be oriented automatically to follow the camera position. In Virtools, you can draw images or videos on 3D Sprites. You can improve the performance of your application significantly by using 3D Sprites especially when you have many repeated objects. For example, you may want to use 3D Sprites to represent trees or cloud of people instead of using actual 3D models of them.

### **Sharing a Mesh by Repeated Objects**

Repeated objects can be duplicated and positioned directly in Virtools. By specifying the option to share the same mesh when you copy objects, you can avoid making duplicated mesh data in the scenes. For example, if you make 200 copies of the same object, there will be only one mesh in the scene. You can reduce the time for mesh processing by rendering engine, and also reduce the size of your composition. You may need to copy objects and place it manually, but you can also write some simple scripts (combination of building blocks) to do it automatically if there is a pattern for the object positions.

### **Collision Detection**

It is recommended to use collision detection techniques to prevent cameras going into certain area or penetrating through walls or objects. Collision detection can take long time and hence affect the performance of your application if the objects' geometries are too complicated. Please keep in mind that you don't need to use actual geometry of the objects, but can use simplify phantom objects for collision detection. For example, if you want to avoid penetrating a statue, you can use a box or a cylinder for collision detection instead of using actual statue object. You can simply place those phantom objects at the same position as actual objects, and hide them to make them invisible.

### **Using Alpha Channel for Complex Objects**

For complicated geometric objects, we can use Alpha channel of textures to make part of the objects transparent. For example, instead of making complex model for a fence, we can simply use a plane and apply a texture with alpha channel to make it look like a fence.

### **LOD**

Level of Details (LOD) is a collections of algorithms used to speed up rendering process, by removing irrelevant details from drawn objects (like very small faces on distant objects).

### **2.3.2.3 Scenario Optimization**

The following technologies are recommended for optimizing the running speed of Virtools programs.

#### **Place & Portal System**

You can organize your scenes into several “places” and define connection or see through window between places using “portals”. By using portal management system of Virtools, you can optimize your scene by letting Virtools engine automatically select visible places to be rendered according to the active camera position and direction.

#### **Scenes**

You should use “scenes” to organize and manage your scenario if you have several different scenes in your composition and need to switch from one to another. Or divide your scenario into several NMOs each of them containing one scene. Then use “Object Load” BB to load it at runtime.

#### **Dynamic Loading**

To make your application start faster and provide smooth experience, you should use step by step dynamic loading approach. For example, you can start application by showing objects with low resolution while downloading objects with high resolution in background. When the high resolution models are downloaded, you can load it dynamically and hide low resolution objects. Or you can start with showing objects around initial location of the camera, while downloading objects far from the initial camera position.

### **2.3.2.4 Interaction**

#### **Easy Control**

Create a simple and easily play system for user is recommended. You should define your control system not only can play with keyboard and mouse but also can through interface.

You should try to make your contents as easy to operate as possible and avoid adding complex operation or control of your scene. You can use simple 2D interfaces (buttons, etc.), simple mouse & keyboard operation. You should not use mouse right button to interact with your application because right click will show pop-up menu for 3DVIA Player.

#### **Navigation**

To make users easy to play different pavilions, we recommend you to follow the suggestions bellow for navigation control:

Basic rules:

- Implement intuitive navigation control by using only mouse movement, mouse left click, mouse wheel scroll, and arrow keys
- Cameras rotate only around global Y-axis and camera's local X-axis to avoid roll rotation
- Add smooth constraint to avoid up and down camera flips

Recommended Functions:

- Look around:
  - Camera rotation by mouse movement + acceleration by pressing mouse left button while mouse cursor is close to window edges (show indicator by 2D icon while acceleration is available).
  - Automatic progressive return to horizontal camera orientation.
- Click & Go (on floor):

Show 3D indicator while mouse cursor is pointing on clickable floor (hide mouse cursor). When clickable floor is clicked (left button), move camera smoothly to the clicked location.
- Click & Go (hot spot):

Show 3D sprites to indicate hot spots, and show 3D indicator while mouse cursor is pointing on hot spot (hide mouse cursor). When hot spot is clicked (left button), move camera smoothly to predefined position associated with the hot spot, and look at hot spot.
- Navigation help: Show button to show navigation help. Use "X" button on upper right of help to close it.

### **Display Information**

It is recommended to use 2D Frames to display information and use "Show"/"Hide" BBs to show/hide them. You can also use "Navigate" BB to open browser to show more rich information.

### **Links**

If you want to link to other EXPO web page we advise you to use "Navigate" BB to implement it.

### **Note**

In order avoid troubles with graphic accelerator compatibility, it is recommended not to use

shaders in your experience pavilion. If you really want to use shaders, you should add judge function in your scripts using “Get Generic Shader Capabilities” BB which will return the information about shader support on the running environment. Also, you can use fixed pipeline “effects” (can be defined on Material Setup) to add some basic visual effects such as reflections.

### **2.3.2.5 Launching**

You can create a html file and a VMO file to publish your contents from “File/Create Web Page” menu. The html file contains the sample codes to embed your composition on web page. If you have external files such as NMO, images, sounds and videos, then you need to copy those files to the same or proper folder in order to open html and play the composition. In order to support browsers other than Internet Explorer such as Firefox, we recommend not using <Object> methods to embed the VMO file. Instead, you should use < JavaScript >Generate3DLifePlayerHtmlTag to embed it.

Because performance of some client computer is not enough to run some heavy compositions, we recommend you to prohibit Full Screen function by adding “allowFullScreen = false” in your HTML code. For example:

```
<script language="JavaScript">  
    allowFullScreen = false;  
    Generate3DLifePlayerHtmlTag("Virtools.vmo",800,600,"Virtools");  
</script>
```

## 3 Others

### 3.1 Data Loading Prompt Page

When accessing Experience Pavilions, some users may encounter slow visiting speed due to uneven network bandwidth, resulting in long-time inaction of screen images. Users may very likely take the situation as no response from the browser or computer failure and then close the browser.

To avoid such, participants can set data loading prompt pages when developing the Experience Pavilions to inform users of the loading progress or by a pop up of reminder to ensure better user experience.

### 3.2 Design of Interactions

During the stage of pavilion design, besides focusing on visitors' browsing habits and psychology, and planning main roaming routes and information about exhibited items, interactive functions design in the pavilion is crucial for participants to attract visitors for an enhanced experience.

Interactive functions can give visitors vivid experience of exhibited items. For example, visitors can freely browse the inside and outside of exhibited items in all directions, especially structures and details and learn about functions by disassembling or assembling the items or simulating rotary movements. Participants can offer a fun experience to visitors by integrating the exhibition message and concepts

Since database support won't be provided by the Organizer, there will be some restrictions for interactive function designs requiring database support. Participants can develop such functions on their own websites, such as the official Expo websites of participants or other related websites and have a link from the Experiencing pavilions to those websites for users to visit.

