**Article ID**Tip-001

**Article Topic**Website Optimization

**Article Title**Reduce Total Number of Objects per Webpage

# Preface

Whenever a web page is requested from the server, the browser also seeks the embedded objects like style sheets, script files, images, media files etc. and tries to download them along with them. Each download creates a web request.

# Explanation

A web page design always consists of various numbers of embedded objects. But sometimes the good UI comes out to be expensive for the user’s time prospective. A user can wait for a page to be loaded with in an average of 10 sec without feedback (refreshing / retrying). Reducing the number of objects in page can reduce the wait time effectively. Here are a few facts.

1. Average Header Size

Each response consists of a header along with the contents, which is of approximately 256 bytes. Consider there are a number of images which are well optimized to reduce their size, but this header information will also be added to the each image download. For a page containing 100 such images will have to download 25 additional KBs, which is huge even the image size is very less.

1. Round-Trip Latency

Each request has its legacy time of average 0.2 seconds for the completion of a round trip. So with a page having the 100 objects will delay in loading for 20 seconds irrespective of the speed of the internet connection.

1. Packet Loss

There is approximate 0.7 percent of the loss of the data in a packet transferred, which will be requested again from the server. Each request creates its own packet of bytes and will lead in to more loss and recovery cycles.

# How to Resolve

Even if the page is optimized for the size of objects, there are a few other things which play a considerable role in the website optimization. Reducing number of embedded objects is one of them. Here are a few tips how we can achieve that.

1. Stitch Images together (create a sprite).
2. Combine the different style sheets into one on the bases of their media type like screen, print etc.
3. Combine the script files according to their features and functionality.

# Conclusion

There are situations where there it is required to have many number of embedded objects in a page, but still we should try to reduce the number to 20/page and should use the cacheable objects so that they are not requested again and again on every time a page is requested.