



Team System Web Access 2008 Scalability Limits

White Paper

January 2009

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2009 Microsoft Corporation. All rights reserved.

Microsoft, Visual Studio, and Windows either registered trademarks or trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

Contents

White Paper	1
Introduction.....	4
Symptoms	4
Cause.....	4
Limits.....	4
Load Test Results.....	5
Test 1 – Large Data Set.....	5
Test 2 - Minimal Data Set.....	6
Workarounds.....	7
References	8

Introduction

This article explains the scalability limits of Team System Web Access and provides workarounds to improve the total number of concurrent users that Web Access can support. Generally, it is not recommended to have more than 100 concurrent users using a single Web Access instance.

Symptoms

If you are experiencing the following symptoms, they may be the result of scalability issues with Web Access:

- Overall Web Access performance degrades
- It takes a long time to open or save work items
- Users occasionally see “Session Expired” errors in Web Access
- Web Access becomes unresponsive and users see “Service not available” error when they navigate to the Web Access website
- Users occasionally see one of the following exceptions when they navigate to Web Access website:
 - Unable to create WorkItemStore
 - Attempted to read or write protected memory

Cause

Team System Web Access is an ASP.NET Web application that uses the Team Foundation Client Object Model API (TFS OM) to access various components of Team Foundation Server such as Work Item Tracking, Version Control and Team Build.

Web Access instantiates multiple copies of Work Item Tracking objects per user and stores them in memory during the users’ sessions. Each user uses a substantial amount of memory and Web Access will consume all the memory available to it after the total number of active sessions exceeds a certain limit.

On 32 bit operating systems, a single process (such as Web Access) can use up to 2 GB memory, which means the total number of concurrent users will be limited to approximately 100 if each user session consumes about 20 MB memory.

In addition to these limits, a large number of concurrent users might cause lock contention issues with the metadata cache files in certain conditions.

Limits

Technically, the scalability limits of Web Access is determined by the number of total work item objects that can be stored in the ASP.NET cache rather than the number of concurrent users. Based on our tests using a large dataset, Web Access hits the upper memory limits after there are approximately 180 work item objects stored in the Asp.Net cache.

In typical usage scenarios where each user usually works with two work items at a time on average, this translates to approximately 100 concurrent users. This limit is determined by various factors such as:

- Hardware specifications of the server

- The amount of memory available to TSWA is the biggest single factor that determines the number of concurrent users.
- Work Item Type Definition complexity
 - More complex work item type definitions will consume more memory and more CPU cycles for rule processing and form rendering.
- The number of team projects on Team Foundation Server
 - More team projects cause the size of the project metadata to increase which causes performance issues with Team System Web Access.
- The amount of data in Work Item tracking
 - As the amount of data increases, the cached object sizes will increase, causing higher memory consumption. In addition to this, larger data will increase the time that it takes to transfer that data from TSWA to the browser, causing degraded performance. For example, opening a work item that contains dropdown type fields with very large amounts of data will be slower and more memory intensive compared to opening a work item with less data. One of the most common examples of this is the team project's valid user group containing large number of users. In this case, work items with fields that list these users (such as in the *Assigned To* field) will consume more memory and more CPU cycles when working with such work items.

Load Test Results

This section provides the load tests results run using the following configuration to determine the total number of concurrent users Web Access can support.

- Single Server TFS Installation with SQL Server 2005 SP2, TFS 2008 and TSWA 2008
- Operating System: Windows 2008 x86
- Hardware: Intel Core 2 Duo 2.4 GHz, 4 GB memory (1.8 GB available to Web Access)

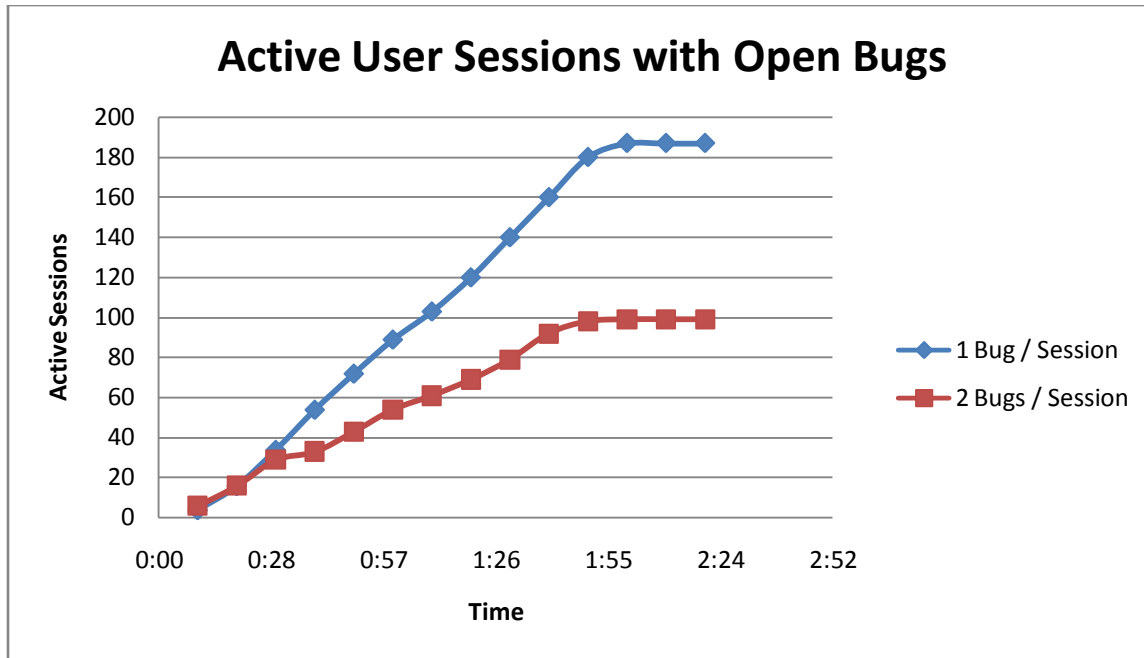
Tests were run in an isolated environment, without any other activity on the server other than the ones required to perform the test. Therefore, these tests don't replicate real-life usage patterns and specifically designed to measure the scalability limits in best case scenarios.

Test 1 – Large Data Set

Using a customized complex Bug work item type with a large data set on the team project, two load test scenarios were run to measure the upper limits for total number of concurrent users. The first scenario opens a single work item for each user, and the second scenario opens two work items.

This test was run against a team project with 5,000 valid project users using a Bug work item type that contains 10 user fields that list these users.

Test results indicate that Web Access starts to hit the upper limits once the total number of work item objects in the memory is around 180. This translates to approximately 100 concurrent users with typical usage patterns.

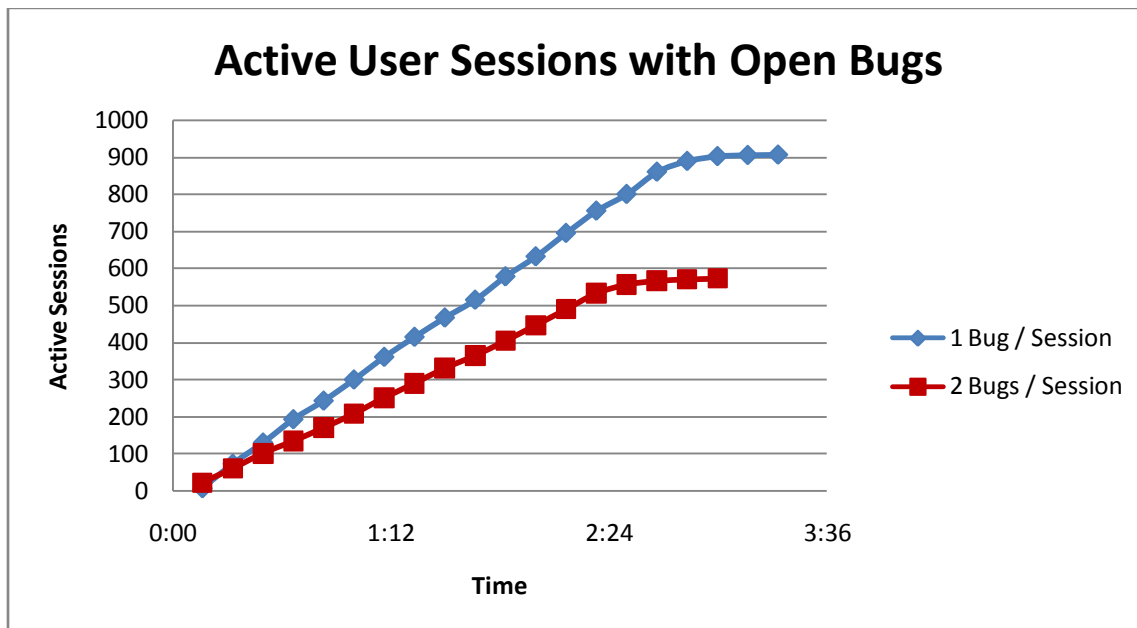


Test 2 - Minimal Data Set

The same test was repeated in the best case scenario, where the simple Bug work item type in MSF Agile process template was used with a very small data set on the team project.

This test was run against a team project with 10 valid project users using a Bug work item type that contains a single user field that list these users.

Test results indicate that Web Access can support up to approximately 500 users.



Workarounds

Install the latest versions of the products

Service Pack 1 includes a couple of performance and scalability improvements. If Web Access is installed on the same server with Team Foundation Server, you must install the TFS, VSTS and Web Access service pack releases together.

- Team Foundation Server 2008 SP1
<http://go.microsoft.com/fwlink/?LinkId=124829>
- Visual Studio Team 2008 SP1
<http://go.microsoft.com/fwlink/?LinkId=122094>
- Team System Web Access 2008 SP1
<http://go.microsoft.com/fwlink/?LinkID=136577>

Increase the amount of memory available to TSWA

The more memory that TSWA can use, the higher the number of concurrent users it can support. On 32 bit systems, a single process (such as Web Access) can use up to 2 GB memory.

Change ASP.NET Cache Memory Settings in Web.config

This change will cause ASP.NET to be more aggressive with reclaiming the memory from the cached objects.

```
<system.web>
  <caching>
    <cache disableMemoryCollection = "false"
           disableExpiration = "false"
           privateBytesLimit = "1000000000"
           percentagePhysicalMemoryUsedLimit = "60"
           privateBytesPollTime = "00:01:00"
    />
  </caching>
```

Use a scheduled task to recycle Web Access IIS AppPool and clear the metadata cache files

This will cause active user sessions to be terminated and the entire metadata cache to be refreshed. It is strongly recommended that you schedule this to run at times when there are no users actively using Web Access.

Metadata cache files are stored in %WebAccessInstallDir%\Cache by default. To learn the actual location, see the WorkItemTrackingCacheRoot entry in web.config:

```
<add key="WorkItemTrackingCacheRoot"
```

```
value="C:\Program Files\Microsoft Visual Studio 2008 Team  
System Web Access\Cache"  
/>
```

A simple batch file is provided below that will restart IIS and delete the metadata cache files that Web Access uses.

```
@setlocal enabledelayedexpansion  
set CACHE_PATH= C:\Program Files\Microsoft Visual Studio 2008  
Team System Web Access\Cache  
IISReset  
del "!CACHE_PATH!" /S /Q
```

Copy the code block above and save it as DeleteTSWACache.cmd, then reference it from the scheduled task to run it every night. Make sure the CACHE_PATH value is pointing to the appropriate directory.

Note that the example uses IISReset command and will restart all the Web sites in IIS. To recycle just the Web Access Web site, use the advanced version of this batch file which is available as a download at:

<http://go.microsoft.com/fwlink/?LinkId=139762>

Install additional Web Access instances

If none of the other workarounds help you improve the Web Access scalability, it is recommended to distribute the load to multiple servers by installing separate Web Access instances on separate servers. Web Access 2008 does not support Network Load Balancing; therefore each Web Access instance will have a separate URL. One option is to install a Web Access instance for each large team so that the team members will always use the server dedicated to them.

References

For more information, please see Visual Studio Team System Web Access on:

<http://msdn.microsoft.com/en-us/tfs2008/bb980951.aspx>