



@ Factory Performance Tuning

Getting Started for @Factory Processing

Table of Contents

Overview.....	1
Preparing for Access Account Visit.....	1
10 Weeks before start of the Service.....	2
8 Weeks before start of the Service.....	3
6 Weeks before Start of the Service	5
3 Weeks before the start of the Service.....	8
2 Weeks before the start of the Service.....	9

Overview

This checklist will help us in preparing your development and Testing environments to be processed by proServices. ProServices will gladly answer any questions on items necessary to most easily and correctly prepare the material to enable a rapid setup of the factory for work on the client's software.

This document outlines the steps **necessary to complete at each time-frame** to prepare for a Performance Tuning service.

Please note the items indicated must be completed (planned / done) at the time frame indicated NOT just getting started. Please prepare to start these tasks earlier as necessary to achieve these activities by the indicated time frame.

Preparing for Access Account Visit

- Purchase Order / Purchase Request / Work Authorization / Contract:**
Legal agreement negotiated / completed.
- Contractor's badges,** or building entry contact and fall-back contact.
- Facilities On-Site:** Office with a telephone and a workstation on the network.
Using temporary cubicle areas.
Note: Delivery Meetings require projection capability and when the presentation is distributed provide for access for the remote teams to view the same information (prior email or conferenced video, or network connectivity prior to the meeting).

10 Weeks before start of the Service

Performance Tuning Service Project Kick-Off Activities

Early stage awareness activities, preparing to contact the “correct” personnel, and matching the executive management expectations with project expectations.

☐ **Project Identification:** _____
Development Manager (per Project): _____
Performance Tuning Contact: _____

☐ **Introduce Service to Project Management / QA:**
Present to the project management the:

- purpose for the service,
- timing of the service,
- high-level impacts on their organization,
- **AREAS OF BENEFIT that will result from the service.**

☐ **Statement of Work Development:** Work with the project manager to prepare a statement of work identifying whether a specific “known” project need drives the **Performance Tuning Service** or whether a general review of the project’s existing Performance Tuning is the business driver. A specific need will often shift the focus of the service from a Broad Based Test Selection Strategy to a narrower Strategy.

☐ **Planning Scope of Tests Selection:** Meet with the project manager and / or QA manager to identify, based on the statement of work, the type of test strategy to employ. (Details on how to plan test selection is given below. Please contact us for any clarifications).

Identifying the tests

Are there any specific performance problems? If so please give the problem description with test case(s) to reproduce the specific problem.

If there are no known performance problems, then select a broad range of tests so that we can look at the whole range of system performance for possible issues.

The performance tuning service reviews the time spent in the code by the tests as they are executed. A common selection of tests for this service is a “reasonable subset” of the Test Plan tests that can “fit the scheduled time”, unless there are specific performance problems. Often Test Plans (TP) are executed over an elapsed multiple months to complete and thus require some reduction in elapsed time for this service. Given that most projects have never had access to Performance Tuning Information as supplied by this service it is more important to get the service run than it is to select the optimal tests.

The following considerations need to be reviewed and understood:

- The selected tests may need to be executed multiple times.
- Data size often slows down the process and can likely be reduced for this service without deterring from the effectiveness of the service. **This reduces the time/effort of running the tests requirements.**
- If there is a specific problem test case given, it will be best to select a test which will not take much time. This can be achieved by stripping down the data, so that only the necessary data remains.

Size of Test Effort Strategy and Considerations

- Consider a subset of the application’s tests that **fits** the following time constraints:
 - Total time on test cycles is expected to be two weeks with variance of +/- one week.
 - Plan for a single cycle of the selected test set of 1 week to allow for running the cycle of tests twice in case of 1st cycle failure.
 - Does not require large amounts of data – can be fast – does not need to be a volume test.
 - We require a **small** portion of test execution
 - Consider a subset of the TP tests that covers as much of the system as possible.
 - In case of test failures we may require identifying the order of tests. We may need to decompose the tests to identify where breakage of the testing process is occurring. In addition it may be necessary to restart testing partially through a test sequence when possible.

Considerations for Performing the Test Effort Strategy

Select one of the following execution strategies:

Tests are Easy for an Outsider (proServices personnel) to pick up and learn to run

- They are well documented
- The tests are scriptable
- The run time environment is general enough to run across the network
- Does not require special hardware access.

Tests are moderately Easy but would need a work around strategy with minor limitations to the above model

- Not easy to pickup and run
- No documentation on how to run the tests, takes a lot of anecdotal or loosely organized information.
- No scripted tests, or not scriptable, requires step by step interactive test execution strategy
- Not a general run time environment, it requires physical hardware access etc.

Tests really need to be run by us (Customer), it would take too much domain expertise to export out the testing techniques.

Based on your selections above, select the corresponding tasks proServices would support:

<input type="checkbox"/> Strategy 1	Plan for proServices personnel to install the test bed (it’s that easy).
<input type="checkbox"/> Strategy 2	Plan for proServices to execute the indicated subset of tests (including 1 st time).
<input type="checkbox"/> Strategy 3	Plan for proServices to execute only subsequent runs of the indicated subset of tests (after teaching them the 1 st time).
<input type="checkbox"/> Strategy 4	Plan for proServices personnel to reinstall the test bed for subsequent test cycles (e.g. ghosting etc. the environment).

8 Weeks before start of the Service

Performance Tuning Service Project Group Planning Preparation Activities

At this stage the project group is gaining initial awareness of the aspects of performing this service that their project will be required to consider. The activities are primarily information communicated to the customer. The Performance Tuning service requires that both “Build Environment Support” and “Run-Time Environment Process Supporting Performance Tuning” are available during the service.

Clarification of directions that the project will need to have decided and planning prepared by the six weeks milestone are communicated and set as clear expectations.

Additional Project Group Contact Identification:

Technical Contact: _____

Build Environment Support Planning – Plan includes access to a complete development environment suitable to build the product executables, including access to build tools and directories. Plan includes preparation for the handoff to Performance Tuning of the instrumented executables produced.

Build Strategies to Select and Plan for by 6 Weeks Prior to Service Start

<input type="checkbox"/> Build Strategy 1	Customer selects the CM version, establishes the setup and build environment (sandbox). Development environment is setup by the customer. Most likely strategy.
<input type="checkbox"/> Build Strategy 2	Customer extract the source base proServices to a delivered setup
<input type="checkbox"/> Build Strategy 3	Customer performs build on proServices behalf.
<input type="checkbox"/> Build Strategy 4	proServices connects into the Customers build and CM environments, selects the version and performs the extraction. We avoid this strategy whenever possible.

Build Scenarios to Select and Plan for by 6 Weeks Prior to Service Start

<input type="checkbox"/> Build Scenario 1	Build Customer Products in Development Environment on a Customer (Shared) Machine. This is a Likely Scenario.
<input type="checkbox"/> Build Scenario 2	Build Customer Products in Development Environment on a Customer (New) Machine. This is a Likely Scenario.
<input type="checkbox"/> Build Scenario 3	Build Customer Products in Development Environment on a Customer (Old) Machine
<input type="checkbox"/> Build Scenario 4	Build Customer Products in Development Environment on a proServices Machine (consider as similar to “new” machine for planning purposes. Requires assembling a build process – makefiles, .dsp files etc. sufficient for remote build.

Run-Time Environment for Performance Tuning Planning – Dialog with the concerned Performance Tuning personnel on the project in order to reduce the impacts on their machine(s) / personnel and so that the work flow is smooth.

Run Time Environment / Test Strategies to Select and Plan for by 6 Weeks Prior to Service Start

Test Strategy 1	ProServices runs the scripted tests and proServices executes tests as many times as needed
Test Strategy 2	Customer executes the tests, proServices collects the instrumentation data
Test Strategy 2	Customer selects a subset of tests from existing test base and execute
Test Strategy 4	Customer executes the tests and send us the instrumented data

Run Time Environment / Test Scenarios to Select and Plan for by 6 Weeks Prior to Service Start

Scenario 1	Running the tests on a Shared Machine at Customer’s site
Scenario 2	Running the tests on a New Machine at Customer’s site
Scenario 3	Running the tests on an old Machine at Customer’s site
Scenario 4	Running the tests on a Machine at proServices site (treat like new machine case).

Type of Test Selection by 6 Weeks Prior to Service Start

Scenario 1	Broad selection of tests across the system. No specific performance problems.
Scenario 2	Narrow range of tests selected. There are specific performance problems identified.

6 Weeks before Start of the Service

Performance Tuning Service Project Group Detailed Planning Activities

Detailed planning needs to be complete at this stage. Review of the strategies and scenarios addressing the approach to manage the expected issues. At this point the Customer should be aware of the implications of issues, any additional blocking issues should have been discussed, and plans including fall-back strategies should be identified for addressing the issues by the two week boundary.

Issues by Scenario - Table to Address by 6 Weeks:

	Scenario	Common Issues Resulting from the Selection of the Scenario
<input type="checkbox"/>	Running the tests on a Shared Machine at the Customer's site	<p>Customer has to Issue Login/Password so that proServices personnel can access customer's machine(s).</p> <p>Advantage is the environment is already set up and the work can be started immediately.</p> <p>Issues include that the Customer still has to identify details of the environment like the operating system, version of operating system, 3rd party software, Hardware requirements etc. (The Impact of this detail is on instrumentation products installation)</p> <p>Issues include the impact on the customer's resources that need to be addressed.</p> <p>Issues include the impact on the customer's time resources that need to be addressed.</p>
<input type="checkbox"/>	Running the tests on a new Machine at the Customer's site	<p>Customer has to Issue Login/Password so that proServices personnel can access customer's machine(s).</p> <p>Advantage is the environment does not impact existing project machine resources.</p> <p>Issues include that the Customer still has to identify details of the environment like the operating system, version of operating system, 3rd party software, Hardware requirements etc. (The Impact is both on getting the build and test environments up and working (eg. Patches needed etc.) and the instrumentation products installation)</p> <p>Issues include whether the correct product (especially infrastructure products like Test control software) have been installed. Address any required machine mirroring software, installation licenses, and Database access issues.</p>
<input type="checkbox"/>	Running the tests on a old Machine at the Customer's site	<p>Customer has to Issue Login/Password so that proServices personnel can access customer's machine(s).</p> <p>Advantage is the environment does not impact existing project machine resources.</p> <p>Issues include that the Customer still has to identify details of the environment like the operating system, version of operating system, 3rd party software, Hardware requirements etc. (The Impact is both on getting the build and test environments up and working (eg. Patches needed etc.) and the instrumentation products installation)</p> <p>Issues include whether the correct product (especially infrastructure products like Test control software) have been installed. Address any required machine mirroring software, installation licenses, and Database access issues.</p> <p>Issues include whether the hardware (CPU, Memory Issues) and OS level of software are adequate for the jobs. These have frequently been neglected and may prove insufficient for the expectation of running the test beds.</p>
<input type="checkbox"/>	Running the tests on a machine at proServices' site.	<p>Run-Time environment would need to be "easy to replicate", not require specialized hardware or software (cost consideration, ease of setup / management considerations).</p> <p>Environment required to run the service has to be provided by the Customer to proServices. In addition this scenario has all of the same issues as servicing a "new" machine scenario at the customer's site.</p>



Build Environment Information Required at 6 Weeks:

Identify the OS Revision (These affect tool revision selection).

Identify the Languages, Compiler & Revision and Build Type (Production, Raw Alpha, Debugging Version) (These affect tool revision, tool configuration selections).



Run-Time / Performance Tuning Environment Information Required at 6 Weeks:

- Plan assembly of all project inventory / material and verify the client's run-time environment will be available as required to run the client's application,

- Plan identification of the subset of tests from the customer's existing test base to be used for execution. Plan for access to the necessary tests required to perform the services. **Note: These tests are a pre-requisite for the service. If the test sets are not available development of the tests must be arranged to supply the tests to perform this Performance Tuning service,**
- Plan for any additional development environment tools (compilers, code generators), libraries or other binaries required to generate the binaries of the product and directory access to the source.
- Plan for any Setup of the run-time environment (machine types, environments, databases, 3rd party products (e.g. websphere etc.), configuration settings etc.),
- Plan for any setup of the Test-Bed including framework, instructions, platform / test bed reset software (e.g. ghosting) and a subset of the desired tests (discussed below) to run the specific area of a product for the analysis to be performed.
- Plan for any specialized run-time environment interfaces (getting binaries into the run-time environment, database schemas, licensing or installation tools (install shield type tools), locations where logs can be written etc.),
- Plan for any Scheduling issues or Machine Availability issues,
- Verify Machine characteristics planning (machine types, disk speed types, network / connectivity capacities, physical memory etc.)



Connectivity / Security / Administration Information Required at 6 Weeks:

- VPN preparation requirements are understood and planned? Access Methodology (telnet, rlogin, remote control)?
- Security impacts are identified, plan access to IP's, Registry of IP's etc.
- Identify process for proServices to add tools to the machine(s) ..permission, availability, licencing (e.g. perl installation, winzip...), and plan their removal at the end of the project.
- Support Plans: Is proServices able to register support issues on machines /databases or receive assistance on them as needed?

3 Weeks before the start of the Service

Performance Tuning Service Connectivity and Access Activities

Access and connectivity are tested and they function as planned to prepare for service level access to the actual inventory later at the two weeks boundary.

- Machines are Installed and Prepared for the Inventory when it is available.**
- Additional Project Group Contact Identification:**
Engineer(s) who can provide answers to questions concerning your environment, your makefiles, and your CM tools.
Found access to General assistance is: _____
Found access to CM assistance is: _____
Found access to SQA Document assistance is: _____
Found access to SW Integration & Test assistance is: _____
Found access to SW Release Identification assistance in: _____
- Login / Passwords tested on each set of machines (build / test) and tools (databases etc.).**
- Security / Integrity of Data Planning Complete:**
Client policies on accounts, passwords, external email between proServices and client, data / software asset movement, planning for cleanup at job completion are prepared.
- Internet / Email / FTP:**
Email Tool Constraints from the Customer:
text only emails, no attachments or other constraints to observe
FTP: Special Access Passwords
Ftp directory location, ftp permissions established
- Machine Config Checked:** _____
Does the machine meet the project hardware configuration requirements. For example on a PC:
Check the machine free disk space (?> 1 Gb) and for sufficient swap size (> 600 Mb).
Check for sufficient memory (>= 128 Mb), Check for sufficient CPU (500 Mhtz).
Screen size > 17" and resolution of 1024x800 or greater.
- OS/ Revision:** _____
- Languages:** _____
- Tools / Service Packs:** _____

2 Weeks before the start of the Service

Performance Tuning Service Inventory Delivery and Service Startup Activities

proServices has access to the Build and Run-Time environments. The customer's machine(s) are implemented as indicated by the above selections of strategy and scenarios. The inventory (source code, build environment, and test environments) are available.

Preparations for the Test Execution and Schedule Access to Personnel are in Place

ProServices Verification of Material Completeness / Service Initiation Status

States include: A=Accessed but not verified
V=Verified, Service Completeness Checking In Progress
I=Insufficient Material, Request for Additional Material Pending
R=Reviewed BOM, Complete Service Initiation Ready

Build is performed to Verify Results (prior to start of service)

Build "works", produces validated results before service begins to instrument.

Interactive Support Is Functioning for Required Services Tool

Performance Tuning Tool passes its self-tests. The project(s) data should be correctly loadable in the tool.

Information Is Available

List of directories of interest machine/path locations _____
home directory: _____
ClearCase or CM tool configuration files _____
Project Files if Needed _____
Source tree path and disk mounts required _____
Tools installation locations _____

How to Build Instructions _____

DSW's list to build, .bat script to invoke, make command line etc.
Don't forget environment variables required, products required.
Known compilation errors in resulting build – but still correct.
Known dead code / projects / directories.
Double click batch file – does whole build.

Verifying Build Instructions _____

Set intersection of .o/.obj vs. .c/.cc/.cpp vs those referenced via make / .dsw
In some services proServices verify build without errors.

Build Environment Setup Info

Compiler Internal Flags (cc -v) _____
Makefiles (If generated then gen them)

Build Environment / Makefiles / DSW's _____

Awareness of -I flags not set in makefiles / .dsw and notify proServices

Source Code _____

Include system / compiler / cots header files included.
Icon, com server...

Test Execution _____

How to execute the tests
Arrangements for environment "re-setup"
Identification of Test Success / Failure Process
Test Configuration Options
Pre-Requisites

Guidelines regarding the post Assessment tests for measuring performance Improvement

The process is as follows.

1) Tests are selected for performance testing. Please see the section “Types of tests” below for detailed description of types of tests.

2) The tests are executed with instrumentation. Now proServices post process the tool log files to determine the performance improvements possible. Once the performance improvements are identified proServices incorporates them and do a build once again.

3) The same tests are ran again to measure the improvement.

Questions regarding retest for performance improvement

This is most difficult part of testing , since proServices makes the proposed changes in the code for measuring the performance improvement. Our plan is to incorporate all the changes we are suggesting for the first test run.

Below is our expectation based on the experience we had with our previous customers.

- 1 - 4 Runs - 30 % of the time this will gather all the required data for assessment
- 5 - 10 Runs – 30 % of the time will eliminate the “simple” blocking problems that prevented the factory run from proceeding to the assessment phase.
- 11 - 15 Runs – 30 % of the time will eliminate the “simple” blocking problems
- Last 10 % of the blocking problems often requires up to 25 additional runs – They require special diagnosis

If fails then we analyze the failures and take out specific changes which cause the failures.

In some cases if the above strategy is not successful, then we take out all our changes and start testing by putting the changes one by one.

Important points to keep in mind while running the tests.

1) please execute the tests identically. For example if the tester does an additional operation in one test(like bringing up another component/window), the results are not comparable.

2) Do not close the true time (tool) window that comes along with the application.

Since this application is instrumented with TrueTime to measure the performance improvements, a true time window will pop up each time the application is launched. In the case of applications creating multiple processes, many truetime windows will be created. So minimize the windows and please do not close them.

3) In general, run tests only once in a test run. Please don't repeat the tests in the same test run, as this will produce data for the repeat run also. So please ensure original run and the modified code test run executes

exactly the same way(if you run the tests twice in original do the same in modified test run also).This will produce data that can be compared.

4) If there are error messages in the logs or error message pop-up, stop the test and check the logs for the I noticed some error messages when the Gateway is launched. One of them is the “Stack Modification error”. If you get this error message or any error messages, please close the application and restart it.

5) TrueTime support suggested increasing the stack size to prevent the “Stack modification error”. We would like to finish the sanity tests first and then increase the stack size, if this is a major issue in running the tests.

6) Some tools accumulate data collected during test run. So it will be not be able to identify the data for each test separately. In this case after each test restart the application.

Appendix -1

Types of tests

Please note that we are measuring the difference between the old code and new code performance. So we need to run the tests at least 2 times, one with the original code and one with the changed code and many more in case of failures. Scripted tests will be ideal in this case.

Fully Automated tests

A fully automated tests defined as a test which doesn't require any human interaction, apart from starting the script. The Automated tests starts up the application and the required servers, run the tests, shutdown/close the application and record the results in the appropriate directory. If you have these type of tests proServices or any one should be able to run the tests given the test name and location .

Fully Automated Data Driven tests

The data driven tests are extremely useful in performance tuning exercises. The data driven test is defined as a test which get input from a file and can execute based on the data from the file. . If you have these type of tests proServices or any one should be able to run the tests given the test name and location .The data sets will have to be decided by you.

Manual Testing

A manual test is defines as a test executed by a person. ProServices wont be able to run the tests unless they are well documented and simple to execute.

Partially automated tests.

We define partially automated tests as test which require a persons interaction in any step of the test. This can be either setup, running the actual test , shutdown or collecting the results. Please note in the case the tests should be repeatable.

