STARTEAM® BEST PRACTICES

Part 2 of 4: Deployment guidelines

Contents

1.1 Welcome to the StarTeam best practices series!

Whether you're a StarTeam newcomer or an experienced veteran, welcome! The goal of this document is to bring you tips, techniques, and practices that help you use StarTeam successfully in your environment. The information presented here supplements the standard StarTeam guides and manuals, providing "how to" advice not found elsewhere. These best practices are cultivated from years of experience helping customers get the most from StarTeam. We hope this information helps your organization as well.

This is the second part in our series of best practices documents for StarTeam. In here you will find guidelines for the successful deployment of StarTeam in your environment. StarTeam is capable of supporting from the smallest to some of the world's largest development teams. We are just a little proud to have customers that have deployed and continue to successfully maintain some of the most distributed teams and largest repositories in the world. While not everybody needs to support more than 160 locations with a total of 45,000 users as one of our customers does, we have learned some valuable lessons in doing so and are happy to share them with you.

1.2. Guidelines for deploying StarTeam

This guide discusses high-level options for hardware deployment with StarTeam. Because StarTeam can be used by small teams, enterprise-scale organizations, and everything in between, there are many options for deploying its components that affect performance, scalability, fail-over, and other factors. In this section, we'll just cover the basics. Where available, we'll provide references to other documents that provide more details on specific topics such as minimum hardware requirements, high availability options, and options for distributed teams.

1.3. Why your mileage will vary

The good news is that StarTeam is a rich application that can be used in a variety of ways. The bad news is that this flexibility makes it difficult to predict exactly what hardware configuration is perfect for your organization. Here are the major factors that affect the performance and scalability of a StarTeam configuration:

- **Repository size:** The number of views and items affect the StarTeam Server process's memory usage, database query traffic, and other resource factors more than any other type of data. Other kinds of data such as users, groups, queries, and filters have a lesser effect on resource demand. Simply put, as the repository gets bigger, more demand is placed on server caching and database queries.

- **Concurrent users:** The number of concurrent users during peak periods has a significant effect on a server. Each concurrent user requires a session, which maintains state, generates commands that utilize worker threads, incurs locking, and so forth. The number of defined users is not nearly as important as the number concurrent users during peak periods. If you use a single metric to gauge your repository "size", use concurrent users.

- **StarTeamMPX:** MPX boosts server scalability, so whether or not you deploy it and whether or not clients enable it will affect scalability. MPX Cache Agents not only significantly boost check-out performance for remote users, but they also remove significant traffic from the server. In short, deploying MPX will bolster your configuration's scalability.

- **Bulk applications:** On-line users that utilize a graphical client typically incur low demand on the server. In contrast, bulk applications such as "extractors" for StarTeam Datamart or Borland® Search and "synchronizers" for integrations such as Borland CaliberRM™ or Mercury Quality Center tend to send continuous streams of commands for long durations. A single bulk application can generate demand comparable to 10-20 on-line users.

- **Application complexity:** Due to its customizability, StarTeam allows you to build sophisticated custom forms, add lots of custom fields to artifact types, create custom reports, and so forth. The more sophisticated your usage becomes, the more commands will be generated and the bigger artifacts will get, both of which increase demand.

Consider these factors when deciding the size of your configuration. Because of the unique factors that define your environment, take the deployment suggestions in this section as guidelines only.

1.4. What is a "large" configuration?

The more sophisticated hardware deployments we'll recommend are those to support a large configuration. But what constitutes a "large" configuration?

Many of us StarTeam enthusiasts have all StarTeam components deployed on a single laptop: a StarTeam server, an MPX Message Broker and Cache Agent, a workflow Notification Agent, a Borland Search server, and various StarTeam clients.[1] Moreover, we often run these alongside our IDEs, Office, and other tools. What this should tell you is that StarTeam components use resources proportional to the work they do. Memory, CPU, and other resources are consumed as the number of users and the volume of information grow.

There are no hard rules about what makes a StarTeam configuration small, medium, or large. However, for our purposes, we'll use these definitions based on concurrent users:

1. A small configuration is one that supports no more than 50 concurrent users.

2. A medium configuration is one that supports no more than 100 concurrent users.

3. A large configuration is one that supports 100 concurrent users or more.

Note that we didn't say anything about data volume or the type of users: on-line users or bulk applications. This is because, in our experience, the amount of data managed by a StarTeam configuration (particularly items) tends to grow proportionally with the number of projects and views, which grow in proportion to the team size. Moreover, the ratio of on-line users to bulk applications tends to be roughly the same across organization sizes. The concurrent user count seems to be the best metric for judging configuration size for purposes of deployment planning.

So how big can a configuration get? To date, we regularly see single StarTeam instances with over 500 concurrent users, over 10,000 total "defined" users, over 4,000 views, tens of millions of items, and up to a terabyte of vault data.[2] In extreme cases we've had customers report a 1,000 concurrent connections on a typical work day, supporting teams in over 160 locations. With continuous hardware advances and software improvements, these limits get pushed every year.

1.5. What happens when a configuration gets "too large"?

StarTeam does not place hard limits on any of the factors described so far. Our philosophy (so far) is to allow you to add as many users, projects, items, etc. as you want, relying on hardware to keep up with the increase in demand. But realistically, there are limits to what you can do with hardware. There are costs incurred by increasingly growing configurations, and these costs eventually overtake the advantages of using a single configuration.

If you are the one charged with capacity planning, we want you to know what happens as configurations get "really big". The negative effects you'll see as a configuration grows are:

- **Start-up time:** The StarTeam Server process performs certain maintenance tasks when it starts such as purging aged audit and security records in the database. As the amount of activity and time between restarts increases, these tasks increase the start-up time. Also, start-up time is affected by the number of unique "share trees" due to initial caches built at start-up time. With well-tuned options, even a large server can start in a few minutes, but it can also take up to 15 minutes or more.

- **Memory usage:** The StarTeam Server process's memory usage is affected by several factors such as the total number of items, the server caching option settings, the number of active sessions (concurrent users), the number of active views, and the number of command threads required. Caching options can be used to manage memory usage to a point, but sessions, active views, and other run-time factors dictate a certain amount of memory usage. On a 32-bit Windows platform, the StarTeam Server process is limited to 2GB of virtual memory. If you enable 4GT

[1] Some of us at Borland have multiple releases of these components installed – and sometimes running – all at the same time!

[2] Note that not all of these limits have been reached by the same configuration. Although some customers have 4,000 views, not all are actively used. Our 500 concurrent-users customer had almost all on-line users. Interestingly, however, the amount of data managed by the vault seems to have little effect on performance or scalability.

RAM Tuning[3], this limit can be pushed closer to 3GB. With the current release of StarTeam supporting 64bit operating systems, these limitations have been relaxed significantly.

- **Command size:** Some client requests return a variable response size based on the number of items requested, the number of users or groups defined, the number labels owned by a view, and so forth. Large server configurations can cause certain commands to return large responses, which take longer to transfer, especially on slower networks. Clients will see this as reduced performance for certain operations such as opening a project or a custom form.

1.6. Deploying multiple small configurations on a single machine

For small- to medium-sized server configurations, you can place all StarTeam server components on a single machine. Furthermore, you can also deploy all components for multiple configurations on the same machine depending on the sum of concurrent users of all configurations. This is illustrated in Figure 1, which shows both basic and MPX StarTeam components deployed.

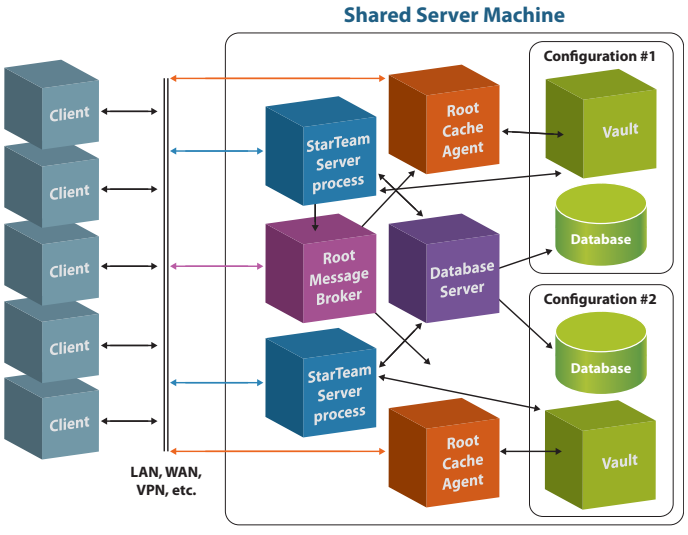

Figure 1: All server components on a shared server machine

You should use a single machine for all StarTeam server components only when the total number of concurrent users for all configurations does not exceed 100. Even though a single configuration can support more than 100 users, each configuration has a certain amount of overhead. Consequently, we recommend that when the total peak concurrent user count reaches 100, it's time to move at least one configuration to its own machine.

With a single machine, all StarTeam Server processes, the root Message Broker, root Cache Agents, and the Database Server process execute on one machine. Here are some rules of thumb for this layout:

- Start with 1 CPU and 1 GB of memory for the database server process.
- Add 1 CPU and 1 GB of memory per StarTeam configuration.
- If you use locally-attached disk for each StarTeam configuration's vault and database partitions, use separate, fast drives to improve concurrency. Also, the disks should be mirrored to prevent a single point of failure. (High availability is discussed in more detail later.)
- If you deploy MPX, all StarTeam configurations can share a single root MPX Message Broker. Though not shown, one or more remote Message brokers may be connected to the root Message Broker.
- If you deploy Cache Agents, each configuration needs its own root Cache Agent, which can share the root Message Broker. Though not shown, one or more remote Cache Agents may be connected to each root Cache Agent.
- Be sure to configure each StarTeam Server, Message Broker, and root Cache Agent process to accept TCP/IP connections on a different port.

Using these guidelines, you can deploy 3-4 StarTeam small configurations on one machine, again only if the total number of concurrent users doesn't peak above 100 or so. Otherwise, the various processes could begin to compete for resources (CPU, memory, disk I/O, and/or network bandwidth), adversely effecting responsiveness. Also, if you start-out with the single-server configuration, don't forget to plan on moving components to their own machines when demand grows over time.

One disadvantage of deploying multiple configurations on a single machine is that they are all affected when the machine must be upgraded, patches need to be installed, someone kicks the power plug, etc.

[3] StarTeam is compatible with the Windows 4GT RAM Tuning option, which boosts the virtual memory limit of a single process on a 32-bit system. See the StarTeam administration documentation for details on using this option.

1.7. Deploying medium configurations

As your configuration size grows beyond what could be called a small configuration, the first thing to move to its own machine is the database process. When you move the database process to its own machine, install a high-speed (1Gbit) dedicated link between the StarTeam server and database machines – we have consistently found that this really makes the separation of the database to its own machine seamless.

Using a separate machine for the database server, multiple StarTeam Server processes and MPX components can still be deployed on the same shared server machine. Because the database processing is offloaded to another machine, the total number of current users can be higher, up to 200-300 or so. A shared database server is shown in Figure 2.

In this diagram, locally-attached disk is assumed for the server and database machines. With multiple configurations, you have multiple vaults and databases,

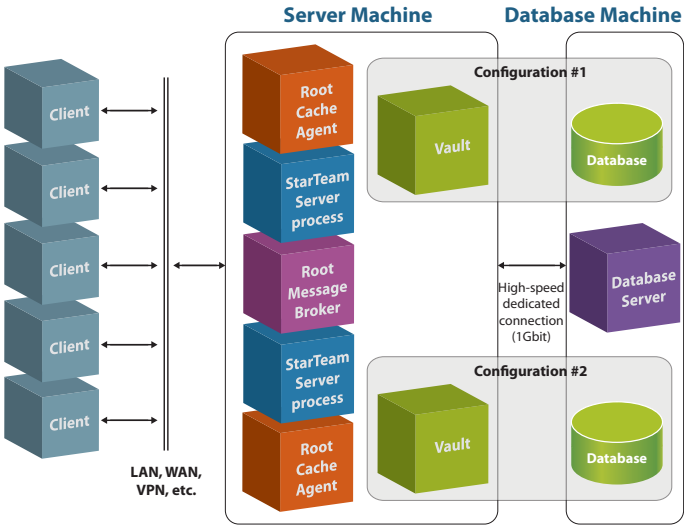

Figure 2: Using a separate database machine

possibly on separate disks. As you consider backup procedures, mirroring for high availability, and other administrative factors, you may find it more cost effective to place all persistent data on a shared disk server (SAN or NFS). Many StarTeam customers with multiple configurations do precisely this. A shared storage server is shown in Figure 3.

Using a shared storage server for all configuration vaults and databases has several advantages. Depending on the storage system, all important data

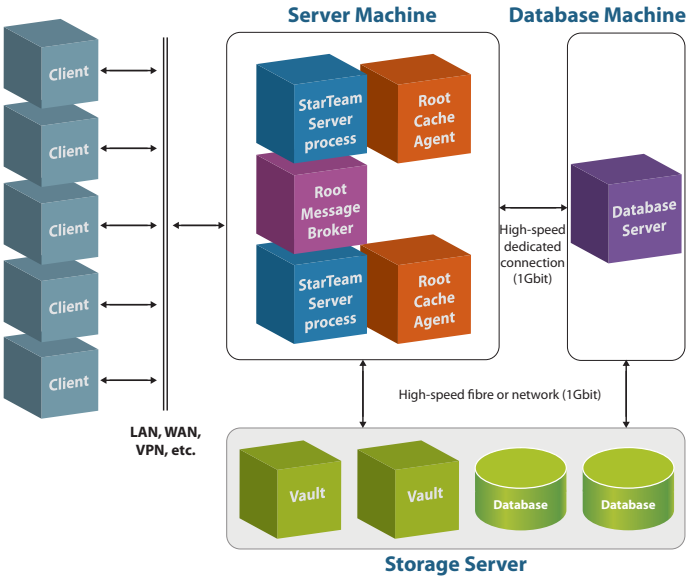

Figure 3: Using a storage server

can be backed-up with a single procedure. Hardware to support mirroring or other RAID configurations can be concentrated in a single place. Many storage systems allow additional disks to be added dynamically or failed disks to be hot-swapped.

1.8. Deploying large configurations

As mentioned before, we consider a "large" configuration one that supports 100 concurrent users or more during peak periods. For these configurations, you should place the StarTeam Server process on its own system. The database process should also execute on its own machine. Though not strictly necessary, the root MPX Message Broker and Cache Agent processes can also benefit by executing on yet another "MPX" machine. Especially when concurrent users rise to 200, 300, or more, moving the MPX processes to their own machine can remove network traffic and other resource contention from the StarTeam Server machine.

A typical deployment of multiple large configurations is shown in Figure 4.

The key points of this multiple, large configuration deployment are summarized below:

- The StarTeam Server process for each configuration executes on its own machine. This is typically a high-end machine with 2 CPUs/2GB of memory for 100-200 concurrent users and 4 CPUs/4GB of memory for 200+ concurrent users. If you expect the user base to grow over time, we recommend you start with the 4CPU/4GB machine.

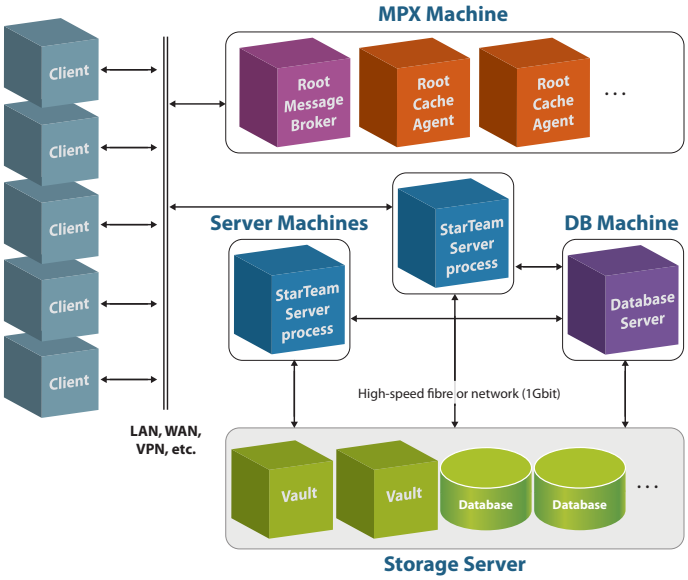

Figure 4: Dedicated StarTeam server machines

- The database server executes on its own machine. Multiple StarTeam configurations can share the same database server. (We've seen up to 8 configurations use the same database server without a performance issue.) Each StarTeam configuration uses its own "schema instance". Each StarTeam server machine should have a high-speed (1Gbit) dedicated connection to the database machine.

- The MPX root Message Broker and root Cache Agents can all execute on a single "MPX machine". Each root Cache Agent requires access to the appropriate vault, but a high-speed dedicated connection is not necessary. File access over the network (e.g., using UNC paths) is sufficient. Note that if you utilize the workflow Notification Agent, you can put it on the MPX machine as well.

- A shared storage server such as a SAN server can be used for all StarTeam vaults and database partitions. Depending on the hardware, an interface (e.g., "host" card) may be needed for each StarTeam server machine in order to access the SAN.

1.9. Clustering for high availability

A graduated series of deployment steps is recommended that improve the availability of a StarTeam configuration. Each step reduces the likelihood of downtime and/or speeds the recovery from a failure, but at additional cost: training, software, or hardware. This approach allows your organization to make a return-on-investment (ROI) calculation that is right for it.

Here, we want to highlight one topic from this paper. StarTeam works with active/passive clustering, in which a "warm standby" node is maintained for quick failover. One general rule to remember is that only one StarTeam Server process can be active for a given configuration at one time. However, StarTeam configuration files can be copied to multiple machines along with all the necessary software. Also, multiple machines under the control of Failure Management Software (FMS) can be connected to the same database (which may be clustered itself), and they can be connected to the same shared storage server for vault access.

Active/passive clustering works like this: the StarTeam Server process on one node in the cluster is started, making it the active node for that configuration. The IP address of the active node is mapped to a virtual "cluster address", which is the address to which clients connect. If the active node fails, the FMS takes care of failover: it starts the StarTeam Server process on a passive machine – making it the active node – and remaps the cluster address to the new active node's IP address. Running clients receive a disconnect message and have to reconnect, but in most cases the failover will occur quickly, so clients can immediately reconnect.

When you have multiple StarTeam configurations, you can "pair" machines so that an active node for one configuration is the passive node for a second configuration and vice versa. Hence, both machines are actively used, and only in a failover scenario one machine must support the processing of both configurations. An example active/passive cluster configuration is shown in Figure 5.

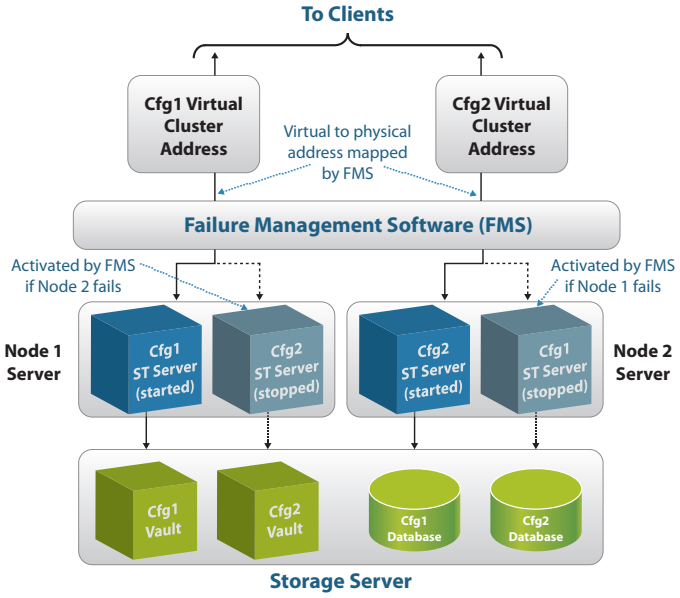

Figure 5: Example active/passive clustering

In this example, the StarTeam configurations Cfg1 and Cfg2 are "paired"; hence one node is active and one node is passive for each one. (The database process is not shown – it might also be deployed on a cluster.)

1.10. Supporting distributed teams

In Part 1, section 1.6, we provided an overview of StarTeamMPX components. When users that access a StarTeam configuration are distributed geographically, MPX becomes essential for ensuring client responsiveness and server scalability. The following provides more details on deploying MPX and other techniques for remote developers. For quick reference, here are the highlights:

- Remote users should enable command compression to reduce the size of client/server messages.
- Remote users should enable both the "StarTeamMPX Server" and "StarTeamMPX Cache Agent" options. The latter should be set to "automatically locate" the closest Cache Agent.
- An MPX Message Broker and Cache Agent should be deployed in each geographic region with more than a few users. The StarTeam administrator should define MPX profiles to make it easy for users to select (once per workstation) the Message Broker that is geographically closest to them.
- Remote Message Brokers should be connected to the root Message Broker using a hub-and-spoke topology. Remote Cache Agents can be configured to auto-locate root Cache Agents as needed.
- Special options may be needed if Message Broker-to-Message Broker or Cache Agent-to-Cache Agent connections will traverse firewalls or VPNs. See the StarTeamMPX Administrator's Guide for details on these options.

Glossary

access control entry (ACE)
A set of access rights (such as the ability to "read", "create" or "delete") granted or denied by an ACL.

access control list (ACL)
A security rule that grants or denies permissions defined in an ACE to a security principal for a specific object. ACLs are defined on objects hierarchically (project -> view -> folder -> item); hence, the lowest-level ACL found in an object's ancestry that matches a given user is the one that applies.

ACID transaction
An update transaction that follows the principles of atomicity, consistency, isolation, and durability.

active/passive clustering
A failover configuration in which a service running on an "active" node has an available "passive" node ready to start and take over in case the active node fails. StarTeam supports active/passive clustering.

artifact
A persistent, stored and versioned object. It is the most important "business object" of VCS and SCM systems.

branch-on-change (BOC)
An item property that controls whether or not the referenced artifact will be branched if an update is made to the artifact through the item.

branch tree
The set of all revision branches for a given artifact. New artifacts start with a main branch with the first revision having version number 0 and dot notation 1.0.

change request
An artifact that represents a defect, an enhancement request, or another reason for a software change.

common ancestor revision (or just common ancestor)
The most recent artifact revision that is common to two artifacts residing in the same branch tree. The common ancestor is used in three-way merge operations.

configuration
An instance of a StarTeam deployment. Each StarTeam

configuration has its own database and vault. At least one project must be added to a configuration before it can be used to hold artifacts.

configuration timestamp
An item property that controls what version of an artifact is referenced. If the configuration timestamp is unused, the item floats to the tip revision of the artifact. Otherwise, it points to revision that was tip as of a specific timestamp.

continuous integration
A software development practice that emphasizes frequent builds and testing. The goal of continuous integration is to encourage the stability of a development stream's current (tip) configuration.

dot notation
A dotted decimal notation assigned to artifact revisions to indicate both the branch on which the revision resides and the relative version number of the revision within the branch (e.g., 1.4 or 1.2.1.5).

exclusive lock
An item "write lock" that can only be held by one user. An exclusive lock notifies other users that the lock owner is potentially modifying the locked item. Compare to shared lock.

file
An artifact that stores properties and content, thereby representing the data held by a disk file.

floating
For items and links, a reference that always points to the current (tip) revision of an artifact is considered "floating". In contrast, a reference can be pinned to a specific revision.

folder
An artifact that represents a disk directory, thereby able to hold other artifacts. In StarTeam, a folder can hold any type of artifact: files, change requests, tasks, etc.

item (or view member)
An association object that connects an artifact to a specific view. In addition to exposing artifacts to views, items facility containment hierarchies and paths, control artifact update behavior, and define "share trees".

label
A named tag that identifies a related set of artifact revisions. In StarTeam, a view label typically identifies a snapshot of an entire view, whereas a revision label identifies a small subset of revisions related to a particular purpose.

link
A connection object that relates two artifacts. Each endpoint of a link can "float" to the current revision of the artifact branch to which it points or be "pinned" to a specific revision on that branch.

main branch
The initial branch created for a new artifact. This branch always has dot notation 1.x; the first revision on the main branch is always 1.0.

pinned
For items and links, a reference that is configured to point to a specific revision is considered "pinned". In contrast, a reference can be floating to the current (tip) revision.

promote
In VCM, a merge operation that propagates changes from a child view to its immediate parent view.

rebase
In VCM, a merge operation that propagates changes from a parent view to one of its immediate child views.

replicate
In VCM, a merge operation that propagates changes from one view to some other view that belongs to the same project.

repository
A centrally-located store for artifacts. With StarTeam, the repository is a database that holds all metadata and data except for file content, which is stored in the vault.

requirement
An artifact that represents a business need. Requirements are early lifecycle artifacts that guide the design, construction, and testing of software components.

revision branch (or just branch)
A stream of artifact revisions that are considered linear, sequential changes. At any given revision, some artifact types can be branched to form a new revision branch.

revision comment
A comment specific to an artifact revision. Compared to other properties, which keep the same value in all revisions until explicitly changed, the revision comment is not carried forward to new revisions and must be set for each change.

revision control
Synonym for version control.

revision tree
Synonym for branch tree.

security principal
A user or group for the purposes of defining security.

shared lock (or nonexclusive lock)
An item "read lock" that can be held by multiple users. Compare to exclusive lock.

software configuration management (SCM)
The process of managing changes to the artifacts in a software development process.

source revision
In a three-way merge, the artifact revision containing changes to be merged to the target revision.

target revision
In a three-way merge, the artifact revision that will be updated with changes contained in the source revision.

task
An artifact that represents a needed, active, or complete unit of work. StarTeam tasks are analogous to tasks in project management systems.

three-way merge
An artifact merge operation in which changes in a source revision are propagated to a target revision. A three-way merge operation considers the common ancestor revision to determine what changes to propagate.

tip revision
The current (most recently created) revision of an artifact. In a rolled-back view (snapshot), the tip revision is the most recent revision available in the snapshot.

topic
An artifact that represents a newsgroup-like message. Topics form conversation threads that increase the collaboration features of a project.

version control
The process of maintaining an accurate history of artifact revisions.

version control system (VCS)
A software application that provides version control features.

version number (or just version)
A unique number assigned to a specific artifact revision. The first revision of an artifact is usually 0 or 1. Each new revision after that is assigned a version number incremented by 1.

view compare/merge (VCM)
A new change management facility introduced in the StarTeam 2006 release. VCM propagates changes a source view to a target view with three types of merge operations: rebase, promote, and replicate.

work record
A record indicating the completion of work on behalf of a specific task.

working folders
A set of folders on a client workstation that are used for file check-in and checkout operations for a specific StarTeam view.

For more information, please visit
borland.com

About Borland

Originating in 1983, Borland is a Micro Focus Ltd brand. Our world class software development products work across the entire Application Development Lifecycle to transform good software into great software. Uniquely, our tools are Open, Agile, and fit for Enterprise.

borland.com

Borland®
A MICRO FOCUS COMPANY

112422_STEAM_BP_2 (04/12)