BitRaider
Digital Distribution

BitRaider Digital Distribution System Technical White Paper - Android™

The BitRaider streaming system provides any software package the ability to be installed and launched with only a fraction of the package's data present locally. The software package itself is not altered in any way and is the same original build produced from the original developer. There are no hooks, ties or any other interfaces to the actual software title.

The BitRaider solution utilizes the Android operating system's underlying file system to provide all of its functionality to the software package. By utilizing this method, it becomes completely invisible to the software package, and as far as the package is concerned, it could just be loading from a slow storage card when it is delivered off the cloud. It is completely unaware that the data may or may not be present on a local storage device when it requests it; requiring no use of virtual disc or virtualization in any way. This solution is quite simple in theory, yet complex in the underlying operation.

The process has a few stages to produce a streaming title out of any software:

1. The title is received by BitRaider and is profiled using the profiling engine. This takes into account many different file formats, compression types, as well as engine type recognition. The profiling engine will then begin sectioning objects out. Once these sections have been identified, it will link up related objects in both time usage and relative location when they may be needed.

2. The title is run 'live' and a differential log is produced. Every time this process is run, the log will be honed tighter and tighter to the actual package objects and their order of usage. This is used to increase accuracy of the package ordering and object usage.

3. The package is produced into its final form and pushed to its final destination which includes uploading to Google Play. This can be hosted on any standard HTTP 1.1 compliant CDN of the customer's choosing. Storage is topology agnostic and can be stored on any mix of web server, FTP, memory stick, DVD, CD or hard drive.

There are multiple methods to deploy the software. In the non-API with no source code required model, the streaming software is incorporated into the device once where the Android operating system can understand and mount the modified OBB package. Installation of titles are handled through the standard APK/OBB model as published by Android. The system uses a simple user mode filter driver and application that only operates then a streaming package is mounted. The BitRaider system is only loaded when the streaming is in use. The security model and data integrity of the installation packages are Android compliant, and meets the security model as published by Android specifications. This allows the software to be operated without any resident or overhead resources in use when not using a streaming or secured package.

The package is installed from either Google Play or another outlet that utilizes the APK/OBB model. This initial install contains all of the information to launch the title. Once the title has started, it retrieves the tables to track the objects, and client information all from the containers on the network and the software launches.

In the API driven model, there is a library that can be used which can be driven from source code to perform the download, update, and launching of the title in a streaming form. This also allows the native title to know the network speeds, objects received, and to make decisions that may change dynamically, or preload anticipated elements from the native title’s server requests.

In the direct source integration model, there is source code and a library that allows the native application to be recompiled with all of the file IO calls being stubbed through the BitRaider library system. This in essence requires minimal to no other source code updating other than adding a header and linking it to the library. From there all calls redirect in an intelligent form that allow the title to become natively streamed.

While the software is in operation, the library is streaming assets into the local machine. It tracks usage in real time as the consumer is using the title. It will pre-cache data that may be needed in the near future based upon where the consumer is and what they are going towards. It will cull out any areas and rebuild this tree on demand multiple times per second if needed. In the event the consumer jumps to an area that is not present, the stub will have the mini-filter hold the title's read call, service the request and then release it back to the title itself as a normal operation.

The system provides many mechanisms to accomplish this task. Each task operates in its own thread to allow multiple operations to occur without congesting one another. At any given time, the system may be transferring data, validating the integrity of the received data, decompressing the network data sent, unpacking the data into the destination files, and performing a final integrity and completion check on the destination files.

As each file is filled, the system will begin to ignore those files that have been completed. As a result, the performance will increasingly mirror the native speeds until ultimately will run exactly the same as the original title.

The system is constantly aware of local resources and network bandwidth requirements. It has the capability to slow down or speed up its own transfer to allow for the title itself to operate correctly online. It also ensures that data is throttled as it is being written to the storage device to prevent an overflow of data, which ultimately would cause the system response to become poor and hamper the experience of the end consumer. The system is also continually throttling its CPU usage and is multi-CPU aware so it operates at the least priority utilizing idle time.

One last component is the patching system. The system allows for titles to be patched in a revolutionary way. When a patch is released for a title, the stub will allow for the title to be started up almost immediately while the patch is being streamed into the containers themselves. This prevents the "download and install" patching mechanisms commonly found in today's software releases.

Additional Functionality of the technology:

- Dynamically detected and loaded international language support to allow full Unicode localization of both interface and application configuration.
- Localized bank switching to allow a single package to have multiple languages, but only need to send the necessary resources.
- Usage and streaming completion data by unique machine / IP identifier.
- Ability to cache and control stored cache content on space constrained devices.

If you have any other questions or inquiries on the BitRaider Distribution system, you can contact us via email at: info@bitraider.com or via telephone at (904) 294-7200.