Loading of Shaped / Compressed Assets

Loading Shaped Assets

Shaped assets are the result of partitioning an asset into several smaller ones, each containing a part of the original asset, based on a predefined set of rules. Please refer to Shape Asset Library chapter for information on how to create them.

The routine for loading shaped assets is similar to the one presented in Loading an Asset section. In this case, Candera::AssetConfig might be used because it allows the definition of multiple instances of Candera::AssetRepository.

The application should derive from the abstract Candera::AssetConfig in order to support specific Candera::AssetRepository instances. The shaped asset repositories need be added, one by one, to Candera::AssetConfig before initializing Candera::AssetProvider.

Loading shaped assets during runtime

There is also the possibility of adding or removing asset repositories even after the initialization of the asset provider. In this case, the application should derive from Candera::DynamicAssetConfig, which allows the asset configurations to be changed during runtime.

To add a repository during runtime, please use Candera::DynamicAssetConfig::AddRepository. In order to remove one, please use Candera::DynamicAssetConfig::RemoveRepository.

Candera::AssetConfigListener::OnAssetRepositoryAdded should be called in the derived classes of Candera::AssetConfig after a new asset has been added, to notify all listeners and load the asset repository. Call Candera::AssetConfigListener::OnAssetRepositoryRemoved when removing an asset repository.

> When dynamically adding of partitioned assets, the newly added assets should belong to the same original asset that the previous loaded are belonging to.

Using Asset Decompression

Candera AssetLoader is able to support and manage asset data compressed by CGI Studio Asset Shaper Tool. Please refer to Compression section for more information related to data compression.

Using of compressed assets inside the application is be possible only if *CANDERA_ASSET_DECOMPRESSION_ENABLED* variable is set to *true* during the build process of Candera. After enabling this feature, *CANDERA_ASSET_DECOMPRESSION_CHUNK_SIZE* will be enabled for specifying the chunk buffer size for decompression.

A *chunk* represents the buffer size that is used for pulling data from the asset to perform the decompression operation, one chunk at a time. Using a smaller buffer size would be time consuming, so a larger buffer size would be recommended in this case, if enough memory is available.

> *CANDERA_ASSET_DECOMPRESSION_CHUNK_SIZE* is relevant only for Candera::FileAssetRepository.

Revision #6
Created 2023-05-01 01:27:02 UTC by Tsuyoshi.Kato
Updated 2024-07-31 23:18:39 UTC by Tsuyoshi.Kato