**2024-06-04: תכנית בניית יישומים – קבצים להכנה, הוראות קינפוג, הוראות הפעלה, והיכן לוג הריצה**

1. **Explanation how to set \_buildCfg.json - general:**
   1. This build-program takes a specified single new version WAB app - "newVersionApp",

Creates many WAB apps, each adapted to a specific customer,

and puts them in the folder path "allAppsFolder".

* 1. This program's folder could be on any disk,so the "newVersionApp", so the folder "allAppsFolder".

Note: **to modify** "newVersionApp", put it in "allAppsFolder" and set "toBuild" as true.

* 1. Next paragraphs elaborate: Preparations before running (2-3), Activation (4), and Logging (5).

1. **Image files to prepare:**
   1. Do insert customers' logo images in the program's sub-folder "**\_appsLogos**".

Each image name should be **identical** to the customer wab app folder name.

Each image extension should be ".png".

1. **How to set config file "\_buildCfg.json":**
   1. At the top, there are 2 "constant" items, pre-adapted to the "newVersionApp". **Don't modify them**.
   2. Specify the next 3 flags:
2. "isDeleteTargetAppOnFail" - Whether to delete target app folder in case of error.

Default value: true (meaning: delete)

1. "isCreateAppByZipElseByCopy" - Whether to create target app folder by zip-unzip or by copy.

Default value: true (meaning: create by zip-unzip)

Note: if by zip-unzip, zip is done once, then all the needed unzips

Note: both unzip and copy preserve files (and directories) modification dates  
 but unzip is faster than copy

1. "isCopyByNodeElseByChildProcess" –

If creating by copy, choose whether to copy by node built-in dir-copy function, or by a PC child process.

Default value: true (meaning: by node)

* 1. "newVersionApp " – This path should point to an existing folder, that could be anywhere.
  2. "allAppsFolder " – The folder for the build program to put the created EXB apps.

It could be anywhere. If it doesn't exist, the program will create it.

for example, if current customers' apps are in "CustomScene" folder,

you can specify a parallel "CustomScene -new" folder, for the program to create and put the new apps there,

then you'll rename "CustomScene" to "CustomScene-old" and "CustomScene-new" to "CustomScene".

Or, you can test app after app in "CustomScene-new" and move the tested apps to "CustomScene".

You can also create the new apps in the current non-empty customer's apps folder if you wish.

* 1. "apps" – an object, containing many "app" objects (for all customer apps that will be created).
     1. For each app object (meaning - for each customer),
* set its object name to the customer WAB app name,
* set its object properties as follows:
  + 1. for each customer there is a **"toBuild"** property,

whose value could be **true** (meaning: to build) or **false** (meaning: to skip).

Note: **to modify** "newVersionApp", put it in "allAppsFolder" and set "toBuild" as true.

* + 1. For customer that does not have **SearchSource2**, set "searchSource2IsAdded" to false.
    2. For customer that does not have an **oblique app**, set this url property to the empty string: "".
    3. For customer that does not have **3dTiles**, set this property either to null or to empty array [].

1. **Activation:**
   1. Double-Click batch file "**\_build.bat**" to run the process in a cmd window,

then examine the results in that cmd window.

* 1. It takes up to one minute to generate each new app. Patience…

1. **Logging:**
   1. A **single** log line is printed to the cmd window while processing each app, displaying:
      1. number of current app versus count of all apps in the config.
      2. Real-Time elapse-time-indicator (in seconds) for each app
      3. Results for each app
   2. The program's sub-folder "**\_log**" contains JSON log files.

Log-file contents is the exact "copy" of the screen, plus a copy of the config.

Log-file name is the run date.