

# **The Ball Content Sync (with Device Client)**

Kalle Launiala, ProtonIT Oy  
[kalle.launiala@protonit.net](mailto:kalle.launiala@protonit.net),  
+358 44 5575665

# Structure of Documentation

- Technical platform/runtime requirements
  - For web developers, back-end developers...
- Identify use-case that answers the following
  - Which group? What to do with them?
  - Data to be synced (mostly DOWN from cloud)
  - Www site template to be synced (UP to cloud)
  - Workspace template to be synced (UP to cloud)
- Workflow for “one-time” setup
- Workflow for using the setup on dev/deploy

# Runtime Environment

Mac, Windows, Linux (= “Mac” Mono)

# Mac OS X (Web Developer)

1. Install Mono runtime MRE (3.2+)
  - a. <http://www.go-mono.com/mono-downloads/download.html>
2. Install latest TheBallDeviceClient
  - a. Download from <https://github.com/abstractiondev/TheBallPlatform/releases/>
  - b. Unpack under home folder ~/Dev/TBDC/
  - c. Verify functionality
    - i. Start Terminal and go to folder “cd ~/Dev/TBDC”
    - ii. “mono ContentSyncTool.exe selfTest”
    - iii. Should give OK for HTTPS test  
Might require “mozroots --import --sync

# Mac OS X (mobile, server developer)

1. Install Xamarin Studio (free for non-Mobile)
  - a. For dependent Mono **install the MDK** version
2. Install NuGet.exe console edition
  - a. <http://docs.nuget.org/docs/start-here/installing-nuget>
3. Grab with terminal (untested)
  - a. `mono --runtime=v4.0.30319 NuGet.exe install TheBallDeviceClient -Prerelease`
  - b. “ContentSyncTool.exe” found under tools-folder
    - i. cd to that folder or include full path below
  - c. “mono ContentSyncTool.exe selfTest”
4. Alternatively get tool from GitHub:
  - a. <https://github.com/abstractiondev/TheBallPlatform/releases/>

# Windows w/ or w/o Visual Studio

- Note! ContentSyncTool.exe is in tools folder
- Visual Studio users grab from NuGet
  - TheBallDeviceClient (prerelease)
  - Recognizes project type (Xamarin vs Windows)
    - Auto-chooses Android vs full from proper folder
- ... or with nuget.exe console
  - <http://docs.nuget.org/docs/start-here/installing-nuget>
  - nuget.exe install TheBallDeviceClient -Prerelease
- For non-VS Xamarin development
  - Android version is under MonoAndroid2.2
    - **Use only on Android** - relies in Android isolation!
  - Full .NET/Mono under “net35” - **config encrypted!**

# Identify Use Cases

Which group, what to do?

# Which group, what they need done?

1. Identify your group - literally which GROUP ID
  - a. <https://test.theball.me/auth/grp/7f05514c-1ad8-4d1c-8d84-f1340927842a/categoriesandcontent/html/group.html>
2. What the group needs to get done?
  - a. It will involve their data (JSON etc) => DOWN
  - b. Workspaces = DEV\_ templates => UP
  - c. Wwwwsite = wwwsite - template => UP



**“One-time” setup syncs**

`“mono ContentSyncTool.exe”`

# Two syncing types options

## Folder-to-folder

- *addSyncFolder*
  - *sync* (to execute)
  - *removeSyncFolder*
- Individual folder trees
- Direction  
UP/DOWN
- Type DEV/wwwsite
  - DEV\_ target prefixed
  - wwwsite dedicated

## Staging-to-Remote

- *setStaging -a, -d*
  - *stgop* (to execute)
    - *--getdata, --putdev, --putlive*
  - *--detachStagingFolder*
- Connection specific staging folder
- Syncs by folder prefix: DEV\_, LIVE\_

# Where to use folder(s) or staging?

- Staging is designed to support deployment control - stage folder can be on TEST server
- Uses sync folder operations under the hood: dynamically sets parameters
  - Is bound to same restrictions: DEV\_ prefix and LIVE\_ only to wwwsite for now
- Folder is suitable for deciding micro-level control on local vs. remote systems
- Example: folder is used to fetch dev/test data to local template folder, staging to publish
  - Use real group data for staging, test data for dev

# What happens in the process?

- Trust is established with group as a whole
  - Sync can upload to either DEV\_ or wwwsite
  - Sync can download from anywhere

So in concrete steps this:

1. Create connection with trust granted
2. Define sync from-to folder(s)
3. Execute “do sync” command to sync them

Same tool can handle any number of connections and syncs: **names** matter

# Sync setup - Trusted Connection

1. Create connection with ContentSyncTool
  - a. “mono ContentSyncTool.exe createConnection”
    - i. -c <give your prefer name>
    - ii. -h <the Ball host; ex. test.theball.me>
    - iii. -g <group ID to connect to>
  - b. Inform the group initiator to approve the connection  
(let him/her know the name you gave to)

# Folder: Sync Folder Definition & use

1. Create sync folder bound to connection
  - a. “mono ContentSynctool.exe addSyncFolder”
    - i. -c <connection name>
    - ii. -s <give sync name>
    - iii. -t <synctype: DEV or wwwsite>
    - iv. -d <syncdirection: UP or DOWN>
    - v. -l <localfolder: not path with ../ notation>
    - vi. -r <remotefolder>
      1. for “UP” and DEV - autoprefixed with DEV\_, for wwwsite/ its always wwwsite
      2. for “DOWN” everything is allowed except for TheBall.CORE/ namespace
2. Use: sync -c <conn name> -s <sync name>

# Staging: Staging definition

1. Choose the local folder to use for staging
2. SetStaging call
  - a. “mono ContentSynctool.exe setStaging”
    - i. -c <connection name>
    - ii. -a <local folder path>
    - iii. -d <data folder name(s) to down sync>
    - iv. To Remove use: --detachStagingFolder
3. Use: stgop [--getdata] [--putdev] [--putlive]
  - a. Any combination of options works in that order

# Running Syncs

Sync what you need IN, sync yours to DEV\_  
or website...



# **“What is different” Syncing**

1. Sync data one folder at a time
  - a. “mono ContentSyncTool.exe sync”
    - i. -c <connection name>
    - ii. -s <sync name>
2. ... or staging parts as a whole with “stgop”
3. Make scripts as clickables to sync all certain kinds of things in a batch
4. Hook scripts to commands in a program

# Results of the sync

- UP copies over to group in URL after groupID
  - /DEV\_<remotefolder>
  - /wwwsite
  - Changes immediately visible - wwwsite needs to be published still with “Publish” to go live-live
- DOWN fetches full folder structure

Allows syncing web developer filesystem 1:1 to what's actually running on the group's content.

# Questions?

Add comments, suggest changes to this doc... contribute :-)