VRD | Advent

∞3.001, 2019/03/23

Contents

Preface

Advent is a project of **VivaRado**, thanks goes to all the contributors, **Google Fonts** and all the people who use it!

Screenshot

Introduction

Advent Pro is a modern display typeface, designed in 2007, this is the new version (#3.001 of 2019) that is delivered as a variable font, along with the classic formats. It supports 14 weights including italics. It is currently maintained and released by VivaRado, by it's original designer Andreas Kalpakidis. What is unique about this version is that thanks to the strict requirements of the Variable Format, all the glyph contours have been reworked, with amazing attention to quality. Then we compressed the kerning to a great degree making the font superbly kerned and compact at the same time.

Contributors:

- VivaRado support@vivarado.com
- Andreas Kalpakides
- Madina Akhmatova
- Dave Crossland
- Michael LaGattuta mjlagattuta@gmail.com
- Behdad Esfahbod

Introduction / Profile

- Company: VivaRado LLP
- Designer: Andreas Kalpakidis
- Twitter: @vivarado
- Google Group:VivaRado Typography Google Group

Introduction / Project Overview

- Project Overview
    - background
    - need
    - scope
    - activities
    - important dates or deadlines
- Project Name: Advent Pro Variable
- Type family name: Advent Pro Var (Advent Professional Variable)
- Proposal Date: 22/08/2018

Introduction / Project Overview / Background

Advent was originally designed during a two year period by Andreas Kalpakidis, in Athens and later in Komotini, Greece during 2006 to 2007. It was an attempt to break some of the Greek script rules and some of the Latin script rules on letterform grouping. All the letters have been in a sense simplified, and Greek Letters where fitted into Latin groups visually. The overall designed was intended to be light, the font is not intended for text but for headlines, logos, and other short message formats as a display font.

Introduction / Project Overview / Need

During 2007 there was a common problem of lack of Greek fonts, even though some agencies took this as an opportunity, Advent was released for free. It was a project that never intended to be something other than an expression of a minimal typographic style.

Introduction / Organisational Documents

Documentation Types:

- HTML - Responsive preview in HTML format - At README directory
- Standard Repository README - At the root of the repository
- PDF - At the root of the repository

Features:

- Responsive Interface
- Synchronized Sidebar
- Hashtag Navigation
- PDF with TOC and Cover

Drawbacks:

- Graphs and Diagrams will not work in github and bitbucket preview, but are still readable.
- Graphs and Diagrams will not work in PDF will be assessed.

Introduction / Phase Introduction

Initiation Phase:

This documentation being part of VivaRado ORGDOC has been implemented over the old advent documentation. Advent itself since 2007 had no real repository or vendor entity other than it was on release by multiple websites for free fonts.

Planning Phase:

All the information on how Advent was updated, produced and offered.

Introduction / Phase Introduction / Initiation Phase

Business Case:

What are the benefits we are trying to get from the project and justification of the decision. It encapsulates the research done to see if the project is worth doing.

- **Initiation Phase** Components ∞0.001:

    1. **Business Case**:
        - Strategic case
        - Management case

Introduction / Phase Introduction / Initiation Phase / Business Case

A business case captures the reasoning for initiating the redesign and variation of Advent.

- **Business Case** Components ∞0.003:
    1. **Solution Options**
    2. **Strategic case**
    3. **Management case**

Introduction / Phase Introduction / Initiation Phase / Business Case / Solution Options

Identified Solution Options:

- Update the old Advent Font to Variable.
- Redesign the old Advent Font to Variable and Include Italics.

Introduction / Phase Introduction / Initiation Phase / Business Case / Strategic case

During the initiation of the redesign of Advent, VivaRado as the current vendor, had to undergo a tool design process, as there was a clear need for faster integration of the new design, we realised that we needed a tool to somehow control the decentralised, but very helpful UFO format.

Introduction / Phase Introduction / Initiation Phase / Business Case / Management case

The management case tests the feasibility of the preferred option, in terms of its deliverability within various tolerances.

Achievability:

By implementing a set of scripts that are now part of VRD TYPL. We could manage a set of UFOs by combining the repeated parts, and keeping them as intact as possible in terms of the UFO format, we introduced internally the EFO as a pseudoformat that allows for implementation of scripts on a whole family of weights. This allowed us to overcome the decentralisation of UFOs and move forward and produce Advent successfully.

Introduction / Phase Introduction / Initiation Phase / Business Case / Management case / Achievements Plan

1. **Milestones**
    1. VRD TYPL
    2. VRD TYPL / TypeFacet Integration
    3. Fontmake Kerning Bug Solution
2. **Dependencies**
    1. UFO
    2. TypeFacet Autokern
    3. Fontmake
3. **Skillset Requirements**
    1. Variable Font comprehension.
    2. Python Programming
    3. Web Application Development
    4. Type Design
    5. Kerning Classification
    6. Kerning Compression

Introduction / Phase Introduction / Initiation Phase / Project Charter

- **Project Charter** Components ∞0.002:
    - Project Introduction
    - Project Goals
    - Deliverables
    - Duration

Introduction / Phase Introduction / Initiation Phase / Project Charter / Project Introduction

- **Project Introduction** Components ∞0.002:
    - Project Name
    - Project Statements
        - Vision Statement
        - Mission Statement
    - Project Definition
        - Problem
        - Opportunity

Introduction / Phase Introduction / Initiation Phase / Project Charter / Project Introduction / Project Statements

Vision Statement:

Contribution to libre great typography.

Mission Statement:

To expand our ideas about typography and contribute to language.

Introduction / Phase Introduction / Initiation Phase / Project Charter / Project Introduction / Project Definition

- **Problem** Components ∞0.001:
    - Problematic Vectors
    - Dirty Contours
    - Old Format
- **Opportunity** Components ∞0.001:
    - New Variable Font Format
    - Cleaner Contrours
    - Compression Concepts
    - Lighter Offering

Introduction / Phase Introduction / Initiation Phase / Project Charter / Project Goals

- Goals for version 3.000:
    - To bring an updated Advent Pro to the Variable format,
    - Add Italics Axis
    - Add Weight Axis
- Goals for version 4.000:
    - Add Cyrillic

Introduction / Phase Introduction / Initiation Phase / Project Charter / Deliverables

- **Deliverables** Components ∞0.001:
    - Advent Pro Variable:
        - Variation Weight Axis
        - Variation Italic Axis
    - Advent Pro 14 Weights:
        - Classic Formats (TTF, OTF, ...)
    - VRD-Typography-Library:
        - EFO to UFOs
        - UFOs to EFO
        - EFO to VAR
        - Kerning Extract Similarity (SIMEX)
        - Kerning Autokern
        - Kerning Compress FlatD
        - Componentize EFO
        - Kerning Adjust UI

Introduction / Phase Introduction / Initiation Phase / Project Charter / Duration

- **Duration** of Advent ∞3.000:
    - 22/08/2018 to 16/02/2019
- **Duration** of Advent ∞4.000:
    - N/A

Introduction / Phase Introduction / Planning Phase

The Planning Phase, is where the project solution is further developed in as much detail as possible and the steps necessary to meet the project's objectives.

The Planning Phase consists of:

1. **Safe Planning**
2. **Stakeholders**
3. **Quality Plan (PQP)**

At this point, the project would have been planned in detail and is ready to be executed.

Introduction / Phase Introduction / Planning Phase / Safe Planning

The project's **Work Planning / Project Plan** is created outlining the activities, tasks, dependencies, and timeframes.

- **Safe Planning** Components (Scope Management):
    - Projeect Plan:
    - Work Planning:
        - Tasks Assignments
        - Dependencies
    - Schedule Planning.
        - Timeline

Introduction / Phase Introduction / Planning Phase / Safe Planning / Project Plan

Decide on the encoding sets and supported language scripts. Decide and plan the weights and how you will generate each weight. Understand the procedures and steps. Calculate or keep track of timelines, steps procedures and pitfalls.

1. **Project Plan** Components ∞0.001:
    1. **Design**
    2. **Production**
    3. **Weights and Masters**
    4. **Kerning**
    5. **Components**
    6. **Language Scripts and Glyph Range**
    7. **Features**
    8. **Delivery**

Introduction / Phase Introduction / Planning Phase / Safe Planning / Project Plan / Design

Advent originally featured 7 weights of geometric sharp curved high rise forms and modernized Greek Letters. This realease features 14 weights, the original updated and improved forms along with italics.

Introduction / Phase Introduction / Planning Phase / Safe Planning / Project Plan / Production

To produce the font, Illustrator and Fontlab was used originally in 2007, updating to Advent Pro Variable in 2018, a set of scripts for Adobe Illustrator (JSX) and Fontlab (Python) where written, additionally python scripts and bash, for the composition and kerning. Later on VRD Typography Library was introduced allowing for easier modifications to the font by utilizing a new format - a container for UFO, called the EFO. VRD TYPL for kerning - and compression and Googles fontmake to compile the final variable font.

Work was done on a Linux box with VirtualBox running Windows 8 and Mac OSX Lion.

Introduction / Phase Introduction / Planning Phase / Safe Planning / Project Plan / Weights and Masters

1. Masters:
    1. Thin (100)
    2. Regular (400)
    3. Bold (700)
    4. Thin Italic (100)
    5. Regular Italic (400)
    6. Bold Italic(700)
2. Master Instances:
    1. **thn** 100 Thin (Hairline) *MM*
    2. **xlg** 200 Extra Light (Ultra Light)
    3. **lgt** 300 Light
    4. **reg** 400 Regular *MM*
    5. **med** 500 Medium
    6. **smb** 600 Semi Bold (Demi Bold)
    7. **bld** 700 Bold *MM*
    8. ~~xbd 800 Extra Bold (Ultra Bold)~~
    9. ~~blk 900 Black (Heavy)~~
    10. **thn*it*** *100 Italic Thin (Hairline) MM*
    11. ***xlg*it** 200 Italic Extra Light (Ultra Light)
    12. **lgt*it*** *300 Italic Light*
    13. ***reg*it** 400 Italic Regular *MM*
    14. **med*it*** *500 Italic Medium*
    15. ***smb*it** 600 Italic Semi Bold (Demi Bold)
    16. **bld*it*** *700 Italic Bold MM*
    17. ~~*xbd*it 800 Extra Bold (Ultra Bold)~~
    18. ~~blk_it 900 Black (Heavy)~~

Introduction / Phase Introduction / Planning Phase / Safe Planning / Project Plan / Kerning

With the help of Typefacet Integrated Autokern, we have obtained the first layer of kerning for the upright bold. By using VRD TYPL Kerning Adjust, we made the corrections, and the rest of the optimisations required per weight.

We have Classified our glyphs in a way where no kerning loss is observed. By dividing by Language Set, without language intrusion between classes. Small Case and Capitals are also non intruding. This increases size minimally but maintains kerning pair loss at zero.

During the process we attempted to maintain the Italics width according to the contour. This created a larger alteration size and jittering italics transition due to changing width - even if the kerning was precise. We eventually opted for the slant-to-right-side-corner and maintained the regular kerning along to the italics and smoother animation on Italics.

Introduction / Phase Introduction / Planning Phase / Safe Planning / Project Plan / Kerning / Standard Sets

All the letter combinations have been kerned but we also perform testing afterwards, for various reasons (Ommited) some kerning pairs are not included. This brings us to testing the kerning on a specific set of letters, the other letters are left to maintain the mechanical, automated kerning.

VivaRado standard kerning sets are defined as follows:

- Letter Based(LB):
    - Latin Capitals(LBLC):
        - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
    - Latin SmallCase(LBLS):
        - a b c d e f g h i j k l m n o p q r s t u v w x y
    - Greek (GREEK UNICODES)(LBGC):
        - Α Β Γ Δ Ε Ζ Η Θ Ι Κ Λ Μ Ν Ξ Ο Π Ρ Σ Τ Υ Φ Χ Ψ Ω
    - Greek SmallCase(LBGS):
        - α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ ς τ υ φ χ ψ ω
    - Numbers(LBNU):
        - 0 1 2 3 4 5 6 7 8 9
- Resulting Permutations that have been Adjusted:
    - Letter Based Permutations (LB):
        - Latin VS Latin Capitals Letter Based Permutation (LBLCLC)
        - Latin VS Latin SmallCase Letter Based Permutation (LBLSLS)
        - Latin Capitals VS Latin SmallCase Letter Based Permutation (LBLCLS)
        - Greek VS Greek Capitals Letter Based Permutation (LBGCGC)
        - Greek VS Greek SmallCase Letter Based Permutation (LBGSGS)
        - Greek Capitals VS Greek SmallCase Letter Based Permutation (LBGCGS)
- Letter to Letter Adjustments:
    - These are small adjustments due to design quirks, and when we decide that a glyph doesn't fit into classes or the class is not satisfying the kerning requirements completely.
- Ommited:
    - Cross Language System Kerning (grek to latn and latn to grek).
    - Greek "sigma1" on the Left Side for all grek.

Introduction / Phase Introduction / Planning Phase / Safe Planning / Project Plan / Kerning / Results

The resulting kerning is:

`{'GG': 4458, 'GL': 761, 'LG': 708, 'LL': 166}`

Total Pairs: `6093`

More information in: Kerning Pair Details

If you notice a possible kerning improvement we would like to hear about it.

Introduction / Phase Introduction / Planning Phase / Safe Planning / Project Plan / Components

Components are created by first running VRD/TYPL/SIMEX to obtain a component similarity index, then VRD/TYPL/COMPONENTS to Componentize the EFO, later you can export to Componentized UFOs.

Introduction / Phase Introduction / Planning Phase / Safe Planning / Project Plan / Scripts

At this moment advent supports Latin, and Greek Encoding.

- **Current Character Support:**
    - *Latin*
    - *Extended Latin*
    - *Greek*
    - *Baltic*
    - *Turkish*
- **Intended Character Support:**
    - The Proposed Encoding/Glyph List: /encoding*list/suggested*encoding.py
    - Current Encoding/Glyph List: /encoding*list/current*encoding.enc

---

Introduction / Phase Introduction / Planning Phase / Safe Planning / Project Plan / Glyphs

The glyph range is ≃ 391 The Proposed glyph list is ≃1500 Glyphs not including the glyphs for Opentype Features (Script Extension).

Introduction / Phase Introduction / Planning Phase / Safe Planning / Project Plan / Features

Advent features at this moment include:

- liga:
    - sub f l by fl;
    - sub f i by fi;
    - sub f f l by ffl;
    - sub f t by ft;
    - sub t t by t_t;
    - sub w w w by w*ww*;
    - sub gamma gamma by gamma_gamma;
    - sub gamma kappa by gamma_kappa;
    - sub lambda lambda by lambda_lambda;

Introduction / Phase Introduction / Planning Phase / Safe Planning / Project Plan / Delivery

Advent Pro Variable will be delivered in 7 Weights & 7 Italic Weights and Variable format. All the Adobe Illustrator scripts, Fontlab Python and additional scripts will be provided. Forks of the original libraries with their alterations, and Encoding Files.

The delivered font files are provided in UFO, WOFF, WOFF2, OTF, TTF, EFO.

All the above files are available here.

Variable font flavors designspace: - Variable with weight axis only - Variable italic with weight axis - Variable with weight and italic axis

---

Introduction / Phase Introduction / Planning Phase / Safe Planning / Project Plan / Usage

To compile from UFO:

```
fontmake -o variable -m '/font.designspace' --output-path '/adventpro-VF.ttf'
```

Or from EFO:

```
python3 '/efo_to_var.py' -s '/font_source/EFO' -o '/adventpro-VF.ttf'
```

Introduction / Phase Introduction / Planning Phase / Safe Planning / Work Planning

Work Breakdown Structure

```
sequenceDiagram
    loop Advent
        LevelA->>LevelB: Advent Work Plan
        alt Glyphs
            alt Design
                LevelB-->LevelC: Cleaning and optimizing
            end
            alt Components
                alt Generate groups.PLIST
                    LevelC->>LevelE: SIMEX to 100%
                    alt Componentize Glyphs
                         LevelD-->LevelE: According to SIMEX 100%
                    end
                    alt Anchoring
                         LevelD-->LevelE: According to Anchors.JSON
                    end
                end
            end
        end
        alt Kern
            LevelB-->LevelC: Flat Kerning
            alt Run Autokern
                LevelC->>LevelE: Autokern Settings
            end
            LevelB-->LevelC: Compressed Kerning
                alt Generate groups.PLIST
                    LevelC->>LevelE: SIMEX to 80%
                    alt Review groups.PLIST
                        LevelD-->LevelE: Best Practices
                        alt Quality Targets
                            LevelD-->LevelE: No Cross Script
                            LevelD-->LevelE: Alpha. Order Parent Children
                        end
                    end
                    alt Compress Flat Kerning
                        LevelD-->LevelE: According to groups.PLIST
                    end
            end
            LevelB-->LevelC: Tested Kerning
            alt Kern Adjust UI
                LevelC->>LevelE: According to LB Permutations
                alt Review, Adjust
                    LevelD-->LevelE: Quality Targets
                    alt Quality Targets
                        LevelD-->LevelE: No Kerning Loss
                    end
                end
                alt Compress Flat Kerning
                    LevelD-->LevelE: According to groups.PLIST
                    LevelD-->LevelE: According to adjustments.JSON
                end
            end
        end
        alt Variable
            LevelC-->LevelD: Export Variable
            alt EFO to VAR
                LevelE->LevelD: End
            else FontMake
                LevelE->>LevelD: By Designspace
            end
        end
    end
```

diagram: #006, ∞0.001, mermaid, Work Breakdown Structure of Advent

Introduction / Phase Introduction / Planning Phase / Safe Planning / Work Planning / Task Assignments

- **Assumptions Research**:
    - VivaRado, Andreas Kalpakidis
- **Docs Update**:
    - VivaRado, Andreas Kalpakidis:
        - Identify Dependencies
        - Identify Resource Requirements
    - VivaRado, Madina Akhmatova
- **Advent**:
    - **Design**
        - VivaRado, Andreas Kalpakidis
    - **Componentization**
        - VivaRado, Andreas Kalpakidis
        - Michael La Gatutta
    - **Kerning**
        - VivaRado, Andreas Kalpakidis:
            - Build
            - Test
        - Michael La Gatutta:
            - Best Practices
- **VRD TYPL**:
    - VivaRado, Andreas Kalpakidis:
        - Build
        - Test
    - VivaRado, Madina Akhmatova:
        - Compression Logic

Introduction / Phase Introduction / Planning Phase / Safe Planning / Work Planning / Dependencies

For the kerning we depend on TypeFacet Autokern.

Introduction / Phase Introduction / Planning Phase / Safe Planning / Schedule Planning

Overview:

At this point we are waiting a PR and in April we can start the planning for Advent 4 that is the inclusion of Cyrillic.

```
gantt
        dateFormat  YYYY-MM-DD
        title Advent Schedule
        section Advent 3
        A:          done,des1, 2018-08-22,2019-02-16
        B:          done,des2, 2019-02-16, 2019-04-01
        A:          des3, after des2, 10d
```

- Completed:
    - ~~**Advent ∞3.000** / July 01 2018 to 2019-02-16~~:
- Current:
    - **Advent PR ∞3.000** / from February 02 2019 to April 01 2019:
        - We are waiting for PR to Google Fonts.
    - **Advent ∞4.000** / After PR in April 2019:
        - Inititation of Planning for the delivery of Advent 4

Introduction / Phase Introduction / Planning Phase / Safe Planning / Schedule Planning / Timeline

- July 01 2018: Start of Redesign
- January 28 2019: Final Kerning for CB for G and L.
- January 31 2019: Contour Fixes, Updates for all weights and anchor alignments.
- February 16 2019: Updated to match contour optimisations of mjlagattuta fork, Updated sources. Observe process at Advent Third Pickup +
- February 17 2019: Updated to match Kerning Classification optimisations of mjlagattuta fork. Further Classification updates, reduction of kerning pairs by 307, Updated sources. Observe process at Advent Third Pickup +
- March 23 2019: Documentation Restructure, according to VivaRado ORGDOC.

Introduction / Phase Introduction / Planning Phase / Stakeholders

We identify the **Stakeholders** by a **Personnel Plan** and create a **Communication Plan** to keep the **Stakeholders** informed.

- **Stakeholders** Components:
    - Applicable Stakeholders (Concious and Unconcious Entities):
        - clients
        - personell
        - funders
        - suppliers
        - equipment
    - Glyph Design Team (Gdes)
    - Kern Testing Team (KeT)
    - Quality Assurance Team (QaT)

Introduction / Phase Introduction / Planning Phase / Stakeholders / Personnel Plan

- Personnel Plan
    - Organizational Structure
        - team members
            - internal
                - Andreas Kalpakidis
                - Madina Akhmatova
            - external
                - Michael La Gatutta
    - Responsibilities and Qualifications
        - Project Management and Accounting: Madina Akhmatova
        - Planning, Development and Design: Andreas Kalpakidis
        - Quality Assurance and Consulting: Michael La Gatutta
        - Acceptance: Dave Crossland

Introduction / Phase Introduction / Planning Phase / Stakeholders / Communication Plan

- Communication Plan
    - Stakeholder Feedback Mechanisms
        - Weekly Notifications
        - VRD Forum
        - Interactive Documentation
    - User Feedback Mechanisms:
        - support@vivarado.com

Introduction / Phase Introduction / Planning Phase / Quality Plan

A **Quality Plan** describes the activities, standards, tools and processes necessary to achieve quality in the delivery of a project.

We can now create a **Quality Plan** by identifying the valid **Quality Targets** we want to achieve. Identify the **Quality Policies** that will be required to achieve them. Identify how to do **Quality Measurement**. Lastly identify how to maintain quality with **Quality Management**.

- **Quality Plan (PQP)** Components:

    1. Quality Targets
    2. Quality Management

Introduction / Phase Introduction / Planning Phase / Quality Plan / Quality Targets

Quality Targets we want to achieve and what are their **Acceptance Criteria**, **Quality Management Procedures**, for each **Applicable Category**

1. Quality Targets Components:
    - Acceptance criteria
        - Glyphs
            - Contour Components
                - Alligned Accents.
                - Components.
            - Contour Quality
                - Point Minimisation.
                - Extremas.
        - Kerning
            - Kerning Loss
                - No Loss on Standard Set permutations.
                - Alpha. Order Parent Children
    - Quality Management procedures
        - Kerning
            - Kerning Loss
                - Kern Adjust Interface Screenshots before and after compression.

Introduction / Phase Introduction / Planning Phase / Quality Plan / Quality Management

Quality Management, the nature of the **Audits**, **Work Verification** by assigning responsible personnel for **Task Fulfillment** and **Task Checking**.

1. Quality Management
    - Audits
    - Tool Scheduling
    - Work Verification
        - Task fulfillment responsible personnel
            - VivaRado
        - Task checking responsible personnel
            - VivaRado and Google Fonts

Glossary

LB: Letter Based, Alphabet / Complete Range.

LBLCLC: Letter Based Latin Capital to Latin Capital

LBLSLS: Letter Based Latin Small Case to Latin Small Case

LBLCLS: Letter Based Latin Capital to Latin Small Case

LBGCGC: Greek VS Greek Capitals Letter Based Permutation

LBGSGS: Greek VS Greek SmallCase Letter Based Permutation

LBGCGS: Greek Capitals VS Greek SmallCase Letter Based Permutation

---

Reference

VRD TYPL/kerning_adjust.py: VRD-Typography-Library-Kerning-Adjust

VRD TYPL/kerning_autokern.py: VRD-Typography-Library-Autokern

charlesmchen TypeFacet Autokern: TypeFacet Autokern
