IBM Advanced Business Partner

INFOSANA

Notes2PDF Converter 1.1 User's Guide

Doc. Rev. 1

Table of Contents

Product summary

Notes2PDF Converter v1.1.0.8 developed by INFOSANA is the tool, which allows you convert your Lotus Notes v7.0.2 documents into PDF format. It's based on Apache FOP project and implemented via additional Java library wrapper together with Notes Java Agent for Lotus Notes specifics. It could run on client side as well as on server in batch mode. The product was validated as “Ready for Lotus Domino software”.

Tests were performed for these configurations:

- Lotus Notes v6.5.5 on WinXP SP2
- Lotus Notes v7.0.2 on WinXP SP2

Main features

- Implemented as Java library (packaged as JAR file) with publicly available API;
- Configurable Page Setup implementation (margins, height/width) ;
- Configurable Header/Footer implementation with Page number/Total number of pages substitution and Lotus RichText fields content;
- Configurable PDF Bookmarks list for multi-document implementation

Installation

Installation packaging

Currently installation is available in form of Windows executable file. It's the same for demo/production versions with the restriction that in demo you have ability to specify only additional Arial font besides default to be used for documents conversion.

Installation steps

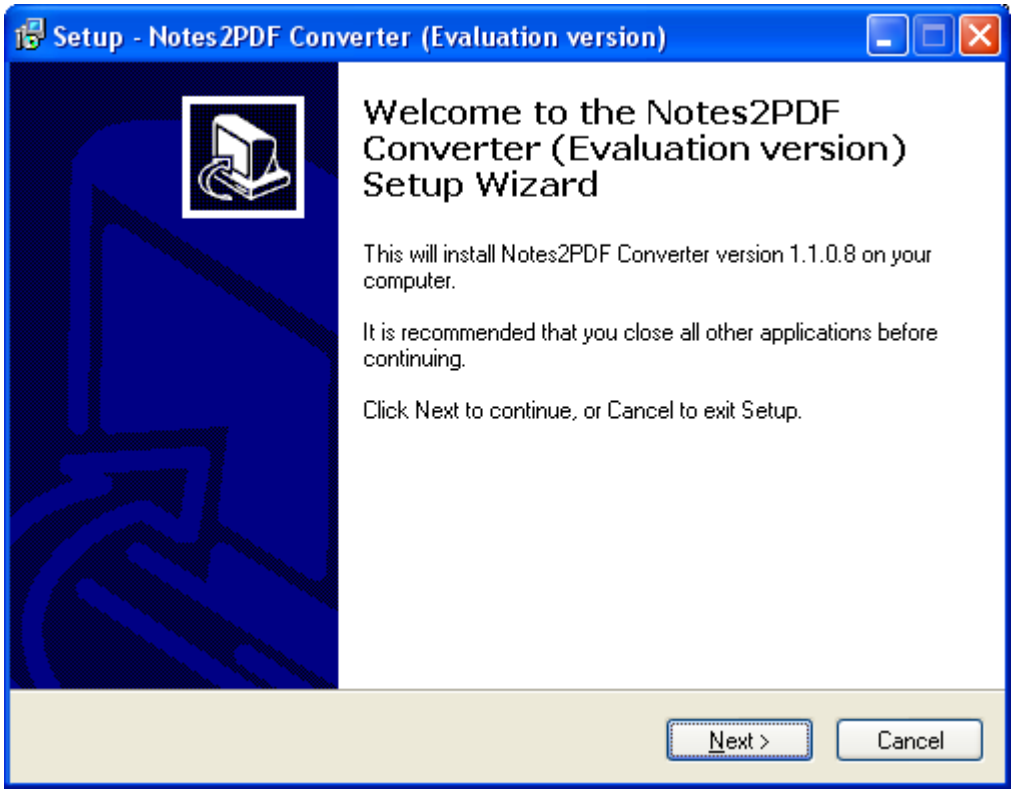

Figure 1. Initial Installation screen.

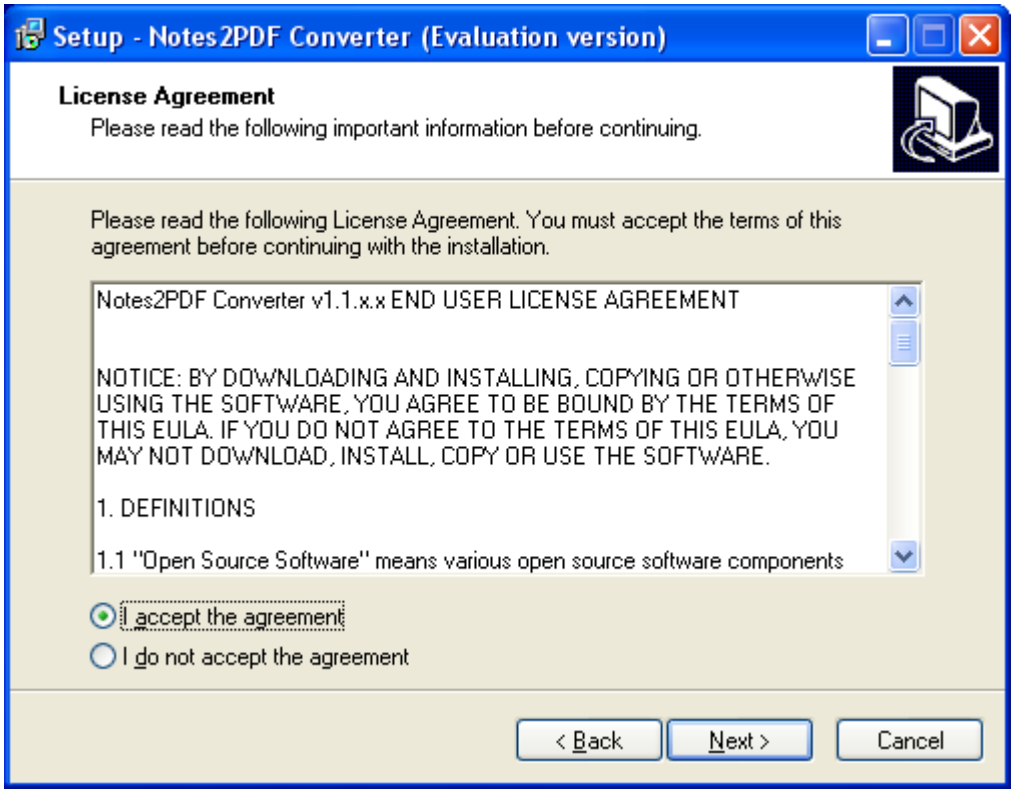

Figure 2. License agreement screen.

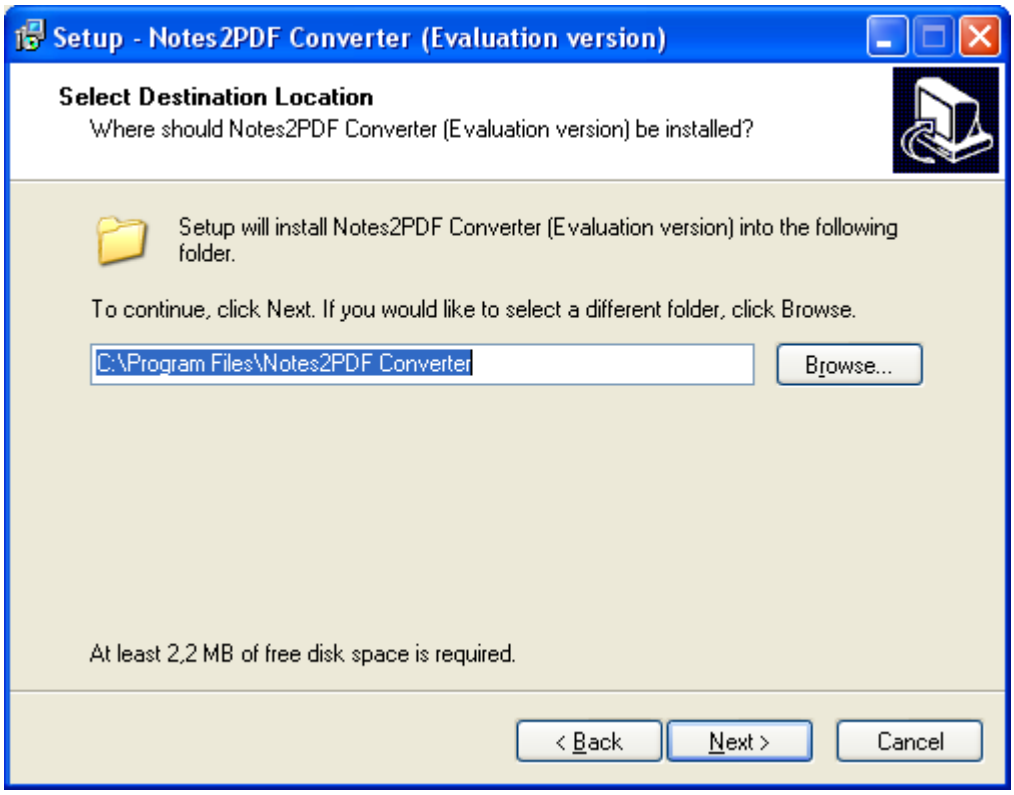

Figure 3. Destination location specification screen.

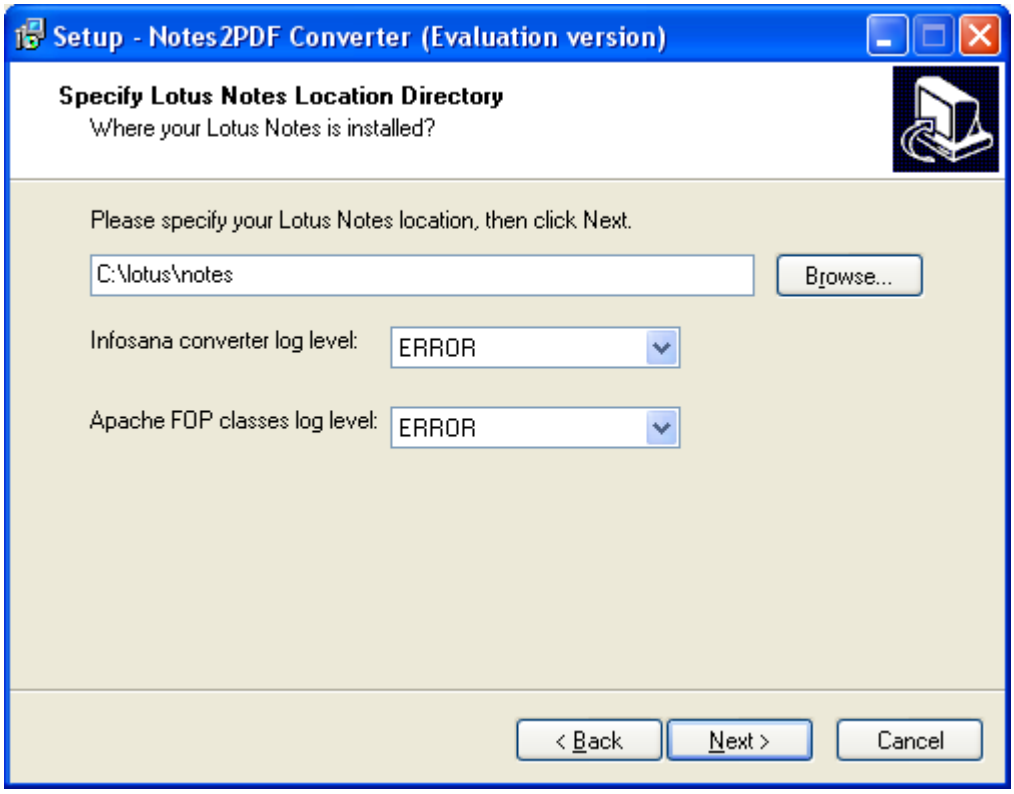

Figure 4. Lotus Notes location and various log levels specification screen.

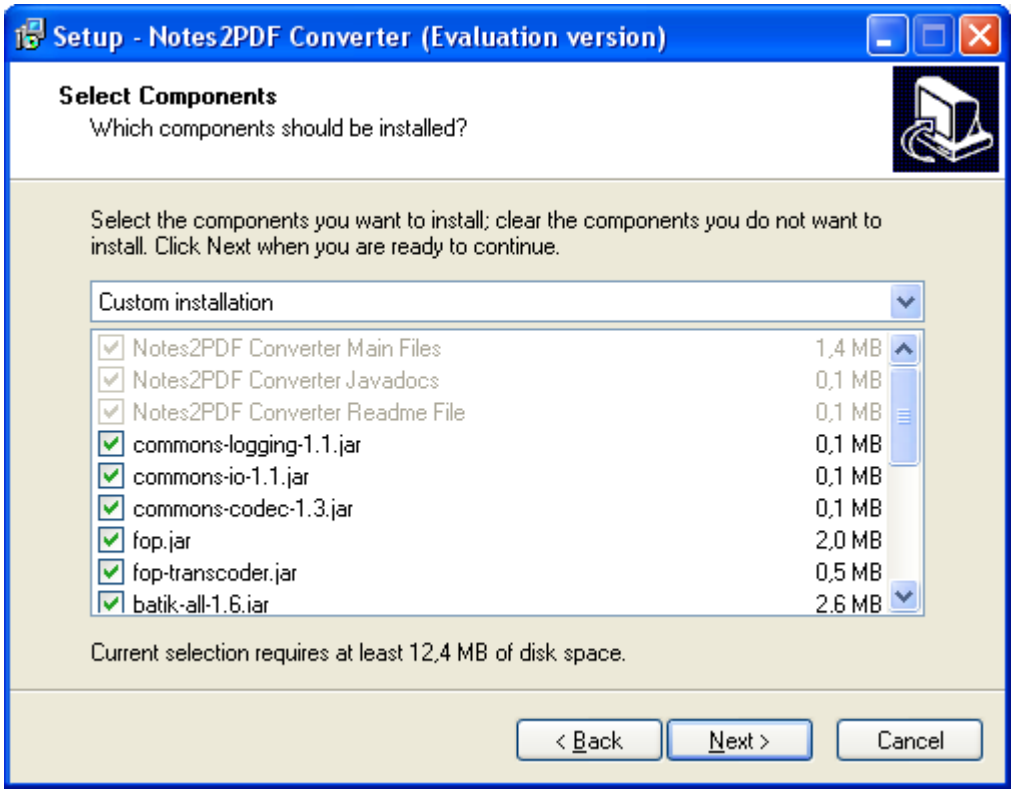

Figure 5. Installation components selection.

“Custom installation” option (see Figure 5) is selected for later Fonts metrics preparation. Those metrics are used during conversion. If that is not required – then “Full Installation” should be selected instead.

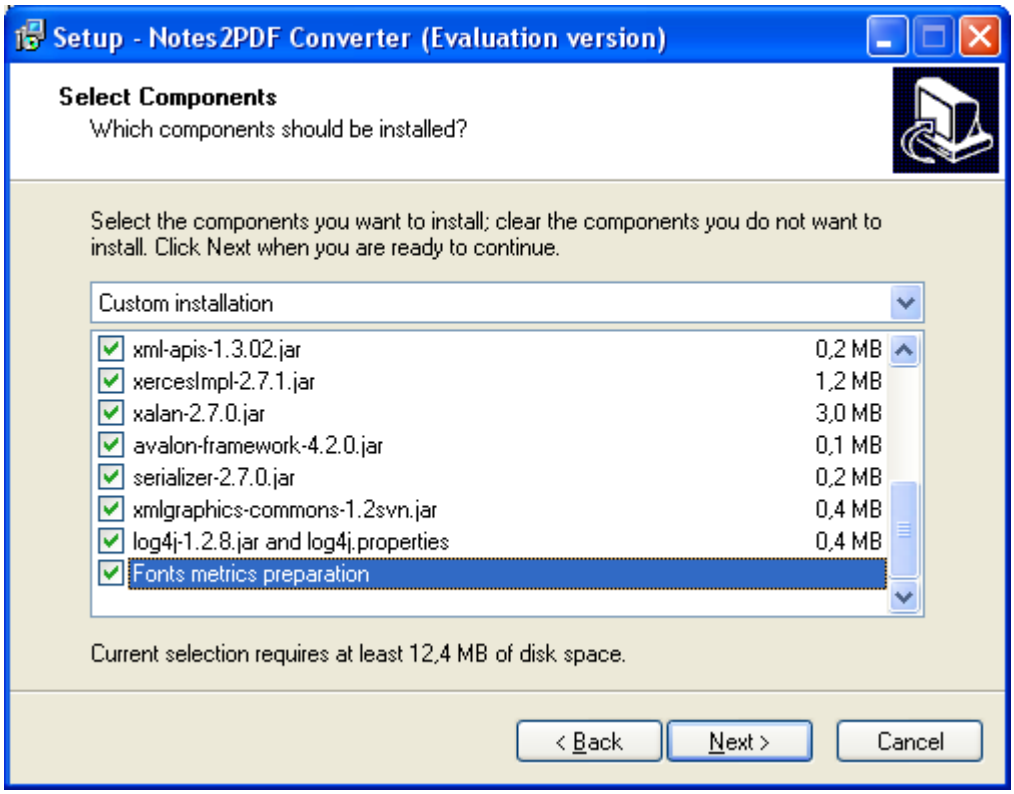

Figure 6. Custom installation with Fonts metrics preparation step selected.

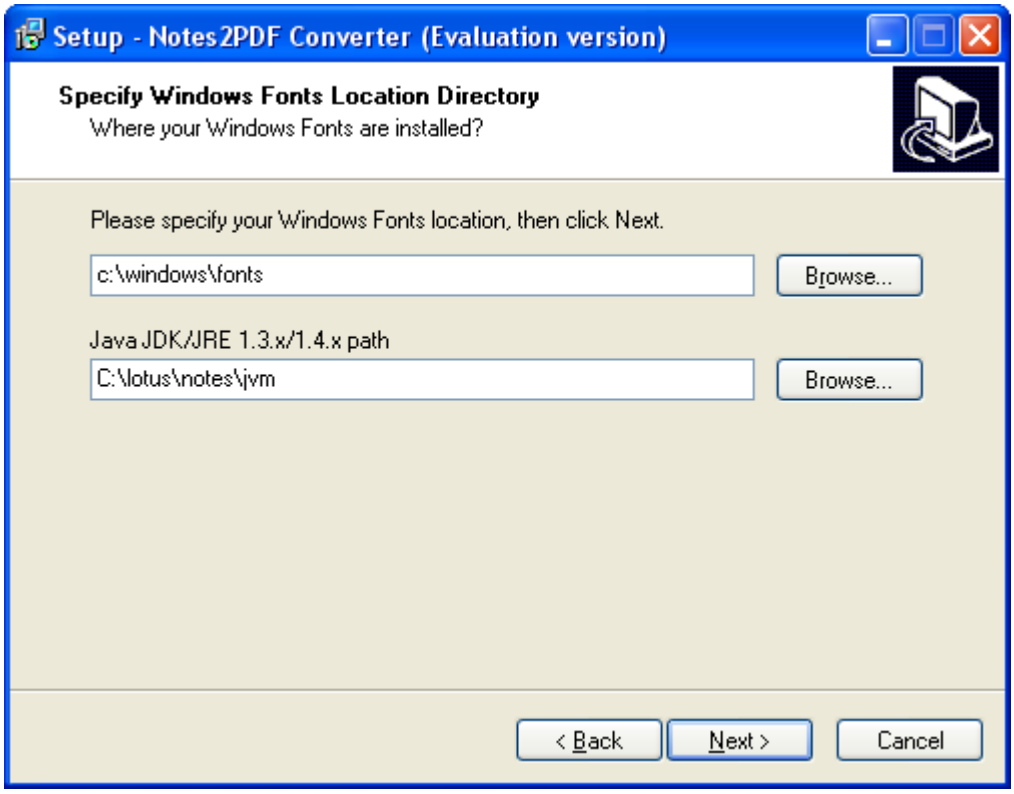

Figure 7. Windows fonts location and JDK path specification screen.

During installation if “Fonts metrics preparation”step is selected (see Figure 6) there will be several fonts related XML files generated in the destination location specified (see Figure 3).

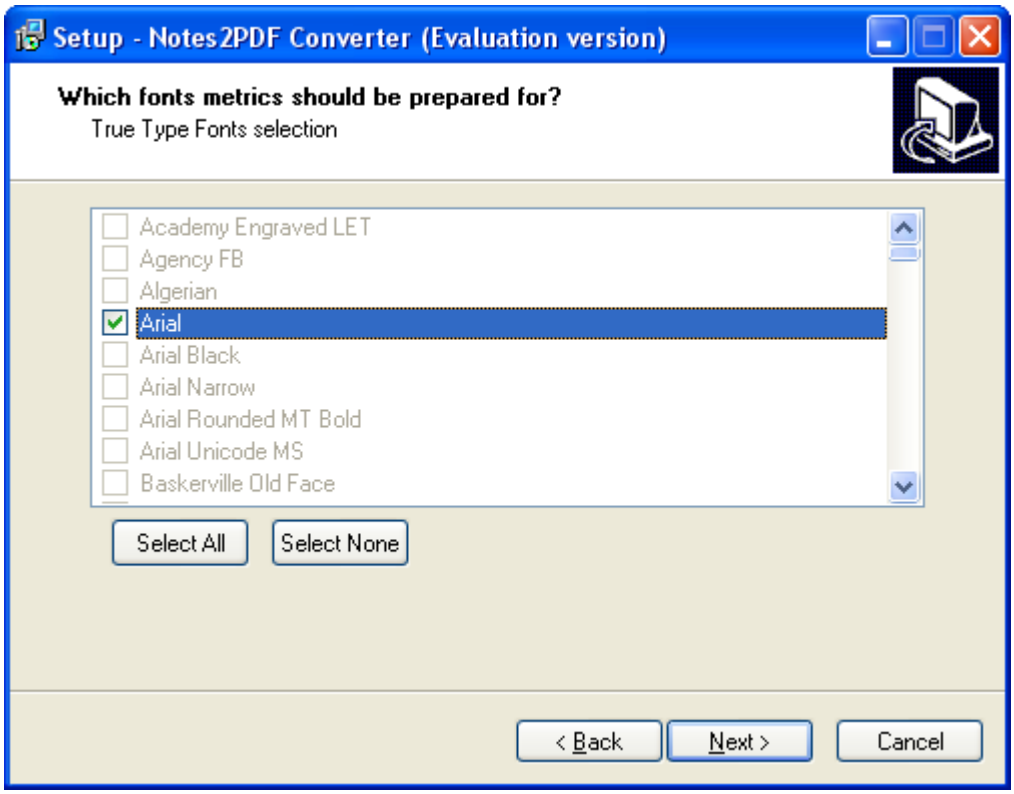

Figure 8. Available Windows fonts selection.

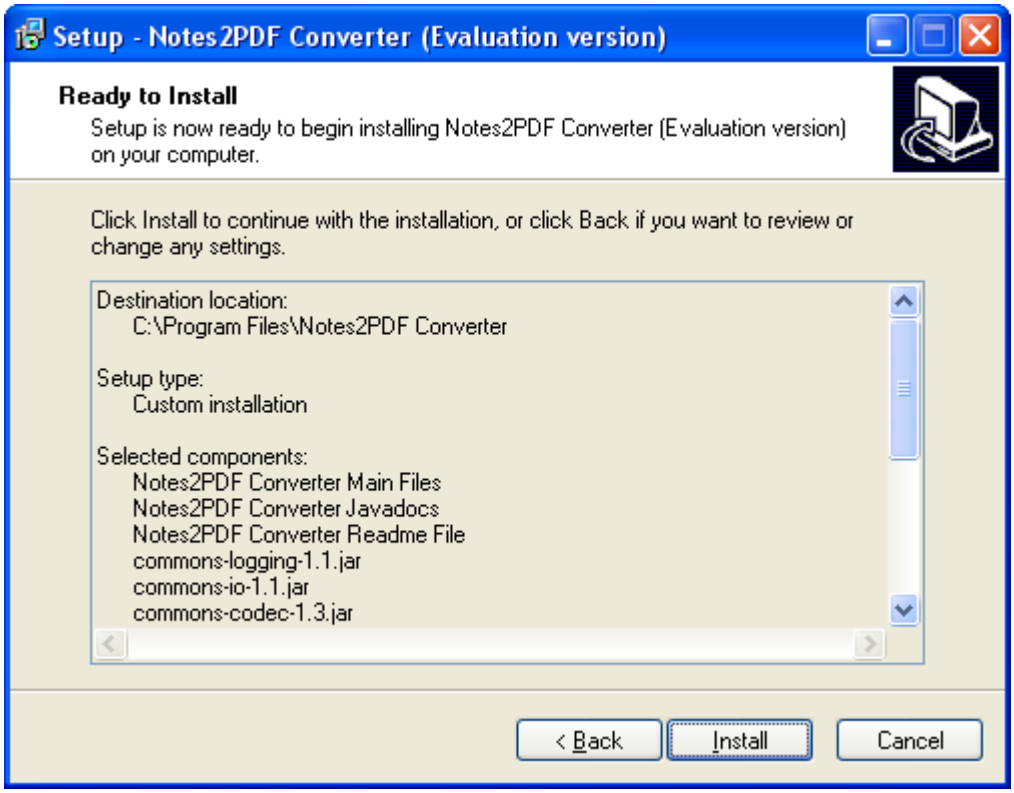

Figure 9. Last screen of selected installation options just before Install launch.

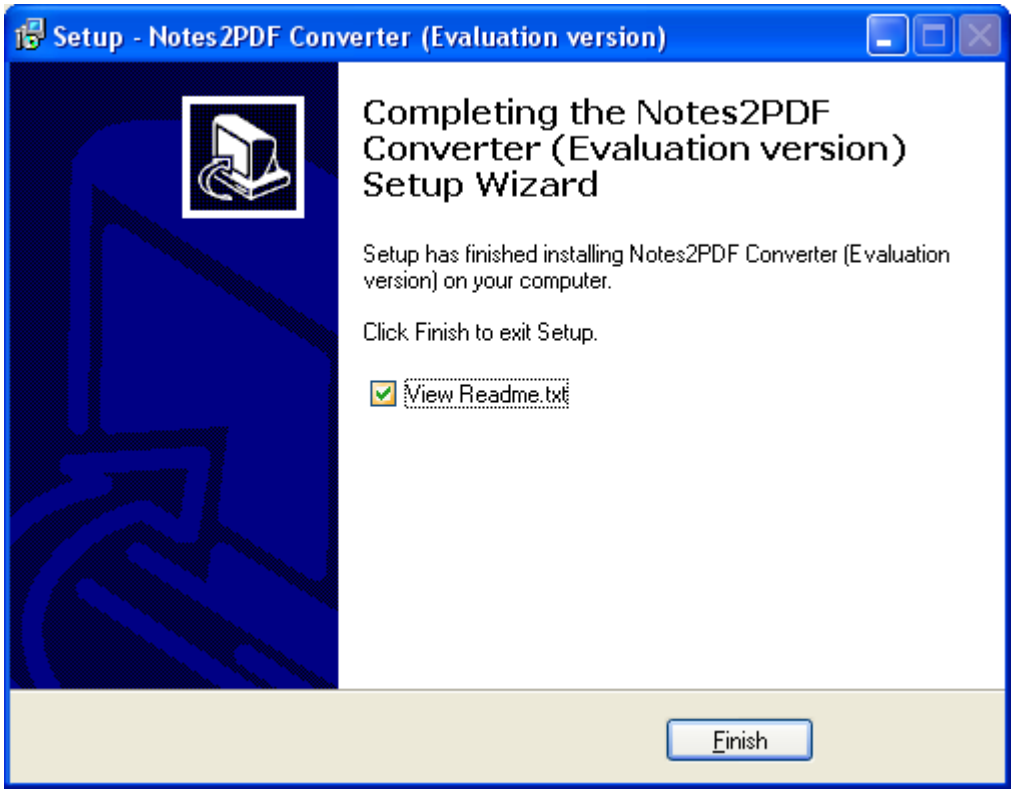

Figure 10. Success installation screen with possibility to view included Readme.txt file.

Directory content after installation

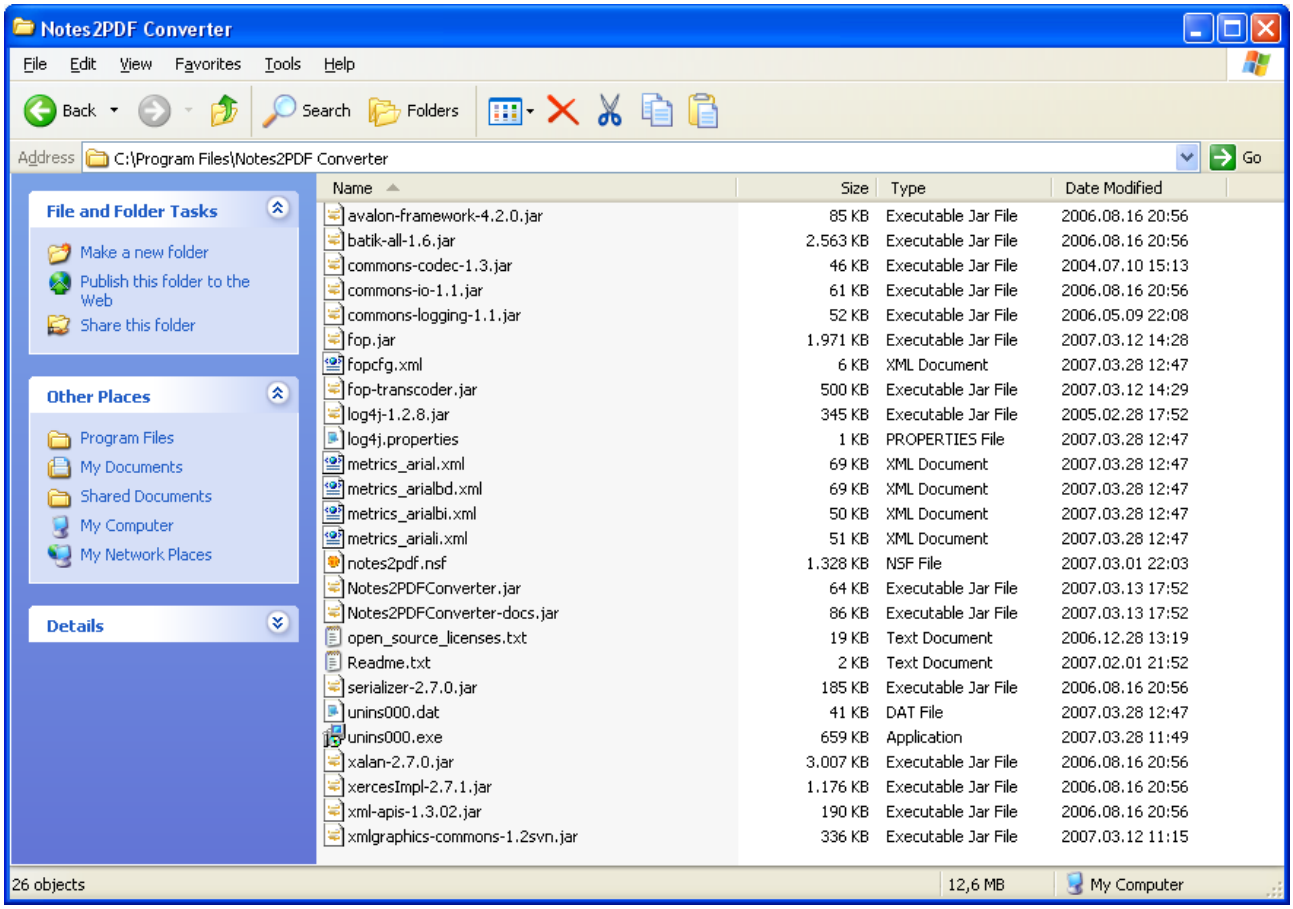

Figure 11. Directory with product's installed files and prepared fonts metrics.

Sample Notes Java agent code

Provided below code demonstrates the abilities of conversion process configuration/invocation via available product's API (copied also during installation as Notes2PDFConverter-docs.jar Javadocs archive). This code is located inside Notes Java Agent called “Notes2PDFAgent v1.1” and is invoked when user press Convert button in “AllDocs” Lotus Notus view in the sample database (notes2pdf.nsf).

```
/**
 * Corresponds to Notes2PDF Converter v1.1.0.0 and higher.
 * Copyright Infosana 2006-2007
 *
*/

import lotus.domino.*;
```

```
import com.infosana.converter.*;

public class JavaAgent extends AgentBase {

      public void NotesMain() {

            try {
                  Session session = getSession();
                  AgentContext agentContext = session.getAgentContext();

                  // (Your code goes here)

                  // Set various page setup properties
                  PageSetup pageSetup = new PageSetup();
                  pageSetup.setHeaderFooterInfo("rtFirstPageHeader",
                        "rtRestPageHeader", "rtFirstPageFooter",
"rtRestPageFooter");
                  pageSetup.setSubstitution(PageSetup.DEFAULT_SUBST_PAGENUMBER,
"##PAGENUMBER##");

      pageSetup.setSubstitution(PageSetup.DEFAULT_SUBST_TOTALPAGENUMBER,
"##TOTALPAGENUMBER##");
                  pageSetup.setUnits(PageSetup.PAGE_UNITS_CM);
                  pageSetup.setLeftMargin(2);
                  pageSetup.setRightMargin(2);

                  // First page configuration
                  pageSetup.setFirstAboveHeader(1);
                  pageSetup.setFirstAboveBody(1);
                  pageSetup.setFirstBelowBody(0.5);
                  pageSetup.setFirstBelowFooter(0.5);
                  pageSetup.setFirstHeaderExtent(2.5);
                  //pageSetup.setFirstFooterExtent(2);

      pageSetup.setFirstPageHeaderAlignment(PageSetup.ALIGNMENT_RIGHT);

      pageSetup.setFirstPageFooterAlignment(PageSetup.ALIGNMENT_LEFT);

                  // Rest pages configuration
                  pageSetup.setRestAboveHeader(1);
```

```
                  pageSetup.setRestAboveBody(0.5);
                  pageSetup.setRestBelowBody(0.5);
                  pageSetup.setRestBelowFooter(0.5);
                  pageSetup.setRestHeaderExtent(0.5);
                  //pageSetup.setRestFooterExtent(2);

      pageSetup.setRestPageHeaderAlignment(PageSetup.ALIGNMENT_RIGHT);

      pageSetup.setRestPageFooterAlignment(PageSetup.ALIGNMENT_LEFT);


                  // Initialize converter with prepared PageSetup object
                  Notes2PDFConverter converter = new
Notes2PDFConverter(pageSetup);
                  converter.setSession(session);
                  converter.setTmpDir("c:/");
                  converter.setPdfFile("test.pdf");
                  converter.setShowProgress(true);
                  converter.setPdfDocFormName("PDFForm");
                  converter.setPdfDocFileFieldName("txtName");
                  converter.setPdfDocRTFieldName("rtFile");
                  String jarFull = session.getEnvironmentString("PDF_CONV",
true);
                  if (jarFull != null) {
                        int pos = jarFull.indexOf("Notes2PDFConverter.jar");
                        converter.setFopConfigFile(jarFull.substring(0, pos) +
"/fopcfg.xml");
                  }
                  DocumentCollection docCollection =
agentContext.getUnprocessedDocuments();
                  System.out.println("Selected count: " +
docCollection.getCount());
                  converter.setDocCollection(docCollection);


                  // Set information needed for bookmarks generation
                  BookmarksInfo bookmarksInfo = new BookmarksInfo("Here we go
collection", "txtBookmarkName");
                  converter.setBookmarksInfo(bookmarksInfo);
                  converter.convert();
            } catch(Exception e) {
                  e.printStackTrace();
            }
```

```
        }
}
```

Contact information

For pricing and licensing please contact sales@infosana.com or call +370 686 09095.