



Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. Adobe Confidential.

The information in this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in this document. The software described in this document is furnished under license and may only be used or copied in accordance with the terms of such license.

Adobe, Adobe InCopy, and Adobe InDesign are trademarks of Adobe Systems Incorporated that may be registered in certain jurisdictions. Macintosh and Apple are registered trademarks, and Mac OS is a trademark of Apple Computer, Inc. Microsoft, Windows, Windows 95, Windows 98, Windows NT and Windows XP are registered trademarks of Microsoft Corporation. All other products or name brands are trademarks of their respective holders.





Goals of the training sessions

- To give an overview of the InDesign CS2/InCopy CS2 SDK.
- To outline some of the new API areas in this version (CS2).
- To briefly discuss porting a plug-in from CS to CS2.

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved.

2



Conventions used in this presentation

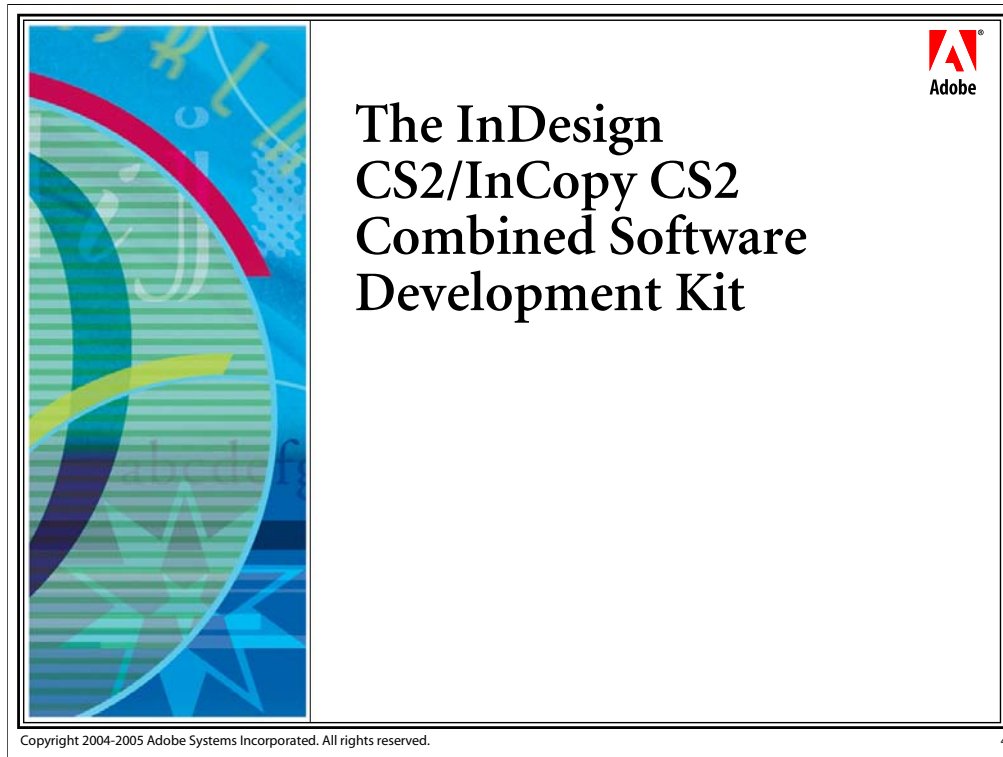
- **File paths**
 - **{SDK}/folder/file.ext**
 - "{...}" denotes some understood, user-specific path
- **URLs**
 - <http://www.mydomain.com/>
- **Document Titles**
 - *Title of a Technical Note*
- **References to an API**
 - `SomeCl assName`

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved.

3

conventions used in the slides.

- File paths are shown in a bold condensed font. {...} denotes some understood path or URL that is user-specific, and is meant to be replaced with the path that applies to your system or context.
- URLs will be in blue.
- Document titles will be underlined and italicized.
- References to an API will be in a monospace font.





Introduction to the Adobe InDesign CS/InCopy CS Combined SDK


- **What can you do with the SDK?**
 - Build and test plug-ins for InDesign CS2/InCopy CS2:
 - **To automate existing application functionality.**
 - To extend application functionality.
 - To integrate workflows involving multiple applications.
 - For all locale versions.
- **For product info**
 - <http://www.adobe.com/products/indesign/>
 - <http://www.adobe.com/products/incopy/>
 - <http://{localized-adobe-domain}/products/indesign/>

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved.


Note that you can use the same SDK to develop plug-ins for all locales. Adobe directly supports InDesign/InCopy CS2 in 13 locales. Middle Eastern and Central European locale versions are also available.

For more product info, please refer to the URLs shown on the slide.

For localized product info, in most cases, you can use the same URL sub-path after the domain. For instance, if you want to look for the Japanese version of InDesign, goto <http://www.adobe.co.jp/products/indesign/>, and for the Italian version, goto <http://www.adobe.it/products/indesign/>. Note the `"/products/indesign/"` part of the URL is common.



Requirements for using the SDK on PC



- **Operating Systems**
 - Windows 2000 + SP 2 or Windows XP + SP 1.
- **Integrated Development Environments**
 - Microsoft Visual C++ .NET 2003.
 - .NET Framework not required for plug-in development.
 - ODFRC compiler supplied by the SDK.
- **Documentation Viewer**
 - A web browser or Windows Help.


Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved.

6


The operating systems requirements for developing with the SDK are identical to the products themselves.

The Integrated Development Environments (IDEs) required for development are shown. In general, we recommend that you use these IDEs.

•For Windows, we require that you use the mentioned IDE. In addition, the Java 2 Runtime and Visual Basic 6 Runtime engines are required for some tools.



Requirements for using the SDK on Mac





- **Operating Systems**
 - Mac OS 10.3 or higher.
- **Integrated Development Environments**
 - XCode 1.1.
 - Metrowerks CodeWarrior Pro 9.2 with a linker patch
 - ODFRC compiler supplied by the SDK.
- **Documentation Viewer**
 - Web browser that supports long Mac filenames, e.g. Safari.

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved.

7

For more information, please read the porting guide.





How to obtain the SDK

- The pre-release SDK is only available to partners who have been invited into the developer pre-release program.
- SDK can be downloaded from:
<https://us.prerelease.adobe.com>
- If you have any questions or issues, contact:
 - Mark Niemann-Ross (mnr@adobe.com)
- The User Forum is the place to discuss the pre-release SDK issues.

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved.

8





Installing the SDK

- **Run the installer from the download**
 - **Mac: Install Adobe InDesign CSx SDK**
 - **Win: Install Adobe InDesign CS SDK.exe**

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved.

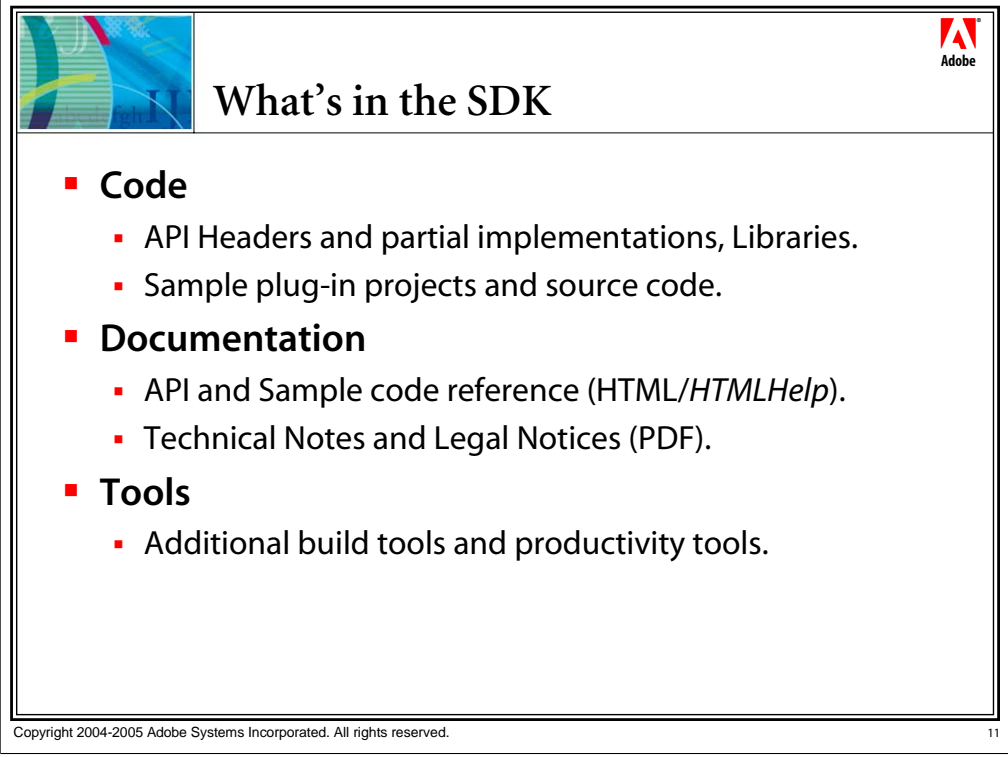
9



What's in the SDK

- **CS2 SDK folder structure is almost identical as the CS SDK except:**
 - The Macintosh SDK frameworks/libraries are moved back in a packagefolder.
 - in CS: libraries were in {SDK}/build/mac/libd|libr.
 - In CS2: libraries and frameworks are in {SDK}/build/mac/debug/packagefolder.
- **Identical footprint between Mac and Win except a couple of platform specific tools.**

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 10



What's in the SDK

- **Code**
 - API Headers and partial implementations, Libraries.
 - Sample plug-in projects and source code.
- **Documentation**
 - API and Sample code reference (*HTML/HTMLHelp*).
 - Technical Notes and Legal Notices (PDF).
- **Tools**
 - Additional build tools and productivity tools.

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 11

The SDK contains three major components.

•Code: comes in two forms:


- API headers and partial class implementations you compile with, as well as libraries you link with
- Sample plug-in projects and source code you can use to learn about the API

•Documentation: also comes in two types:


- API and sample code reference in HTML and HTMLHelp format (for Windows only). The location of these files and instructions for decompressing the HTML tar-ball `{SDK}/docs/references/sdkdocs.tar.gz` is shown in the `{SDK}/Readme.txt` and `ReadMeJ.txt` files. There is also a text file that describes how to turn the HTML-based documentation into an Apple Help bundle for use on the Macintosh.
- Technical notes and legal notices in PDF. We assume you have at least the free Adobe Reader.

•Tools for helping you develop with the SDK:

- IDE plug-ins such as ODFRC and build tools.
- Productivity tools such as GrayGoose and DollyXs.




What's in the SDK: Development Productivity Tools




- **DollyXs**
 - Revamped cross-platform project generator.
- **Graygoose**
 - Mac project converter that convert a CS (CW 8.3 based, PEF format) project mcp to a CW 9.2 project that will build plug-in as package bundle in Mach-O format.
- **DebugWindow (Mac only)**
 - For viewing **TRACE** outputs with Debug build.

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved.

12





What's in the SDK: Documentation



- **Reference Documentation: {SDK}/docs/references**
 - About API classes.
 - About "Boss classes".
 - About SDK Samples.
 - Overview ({SDK}/docs/references/index.html).
 - Explode {SDK}/docs/references/sdkdocs.tar.gz first.
- **Technical Notes: {SDK}/docs/guides**
 - More in-depth coverage about specific topics.
 - See {SDK}/Readme.txt for an overview.



Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 13



How do we get started?

- The best place to start is the *Porting Guide*
 - `{SDK}/docs/guides/portingguide.pdf`
 - How to set up your IDEs.
 - On the Mac: How to create a plug-in project from scratch, or how to convert InDesign/InCopy CS plug-in projects for use with InDesign CS/InCopy CS2.
 - On the PC: The CS guideline is still true for this version.

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 14

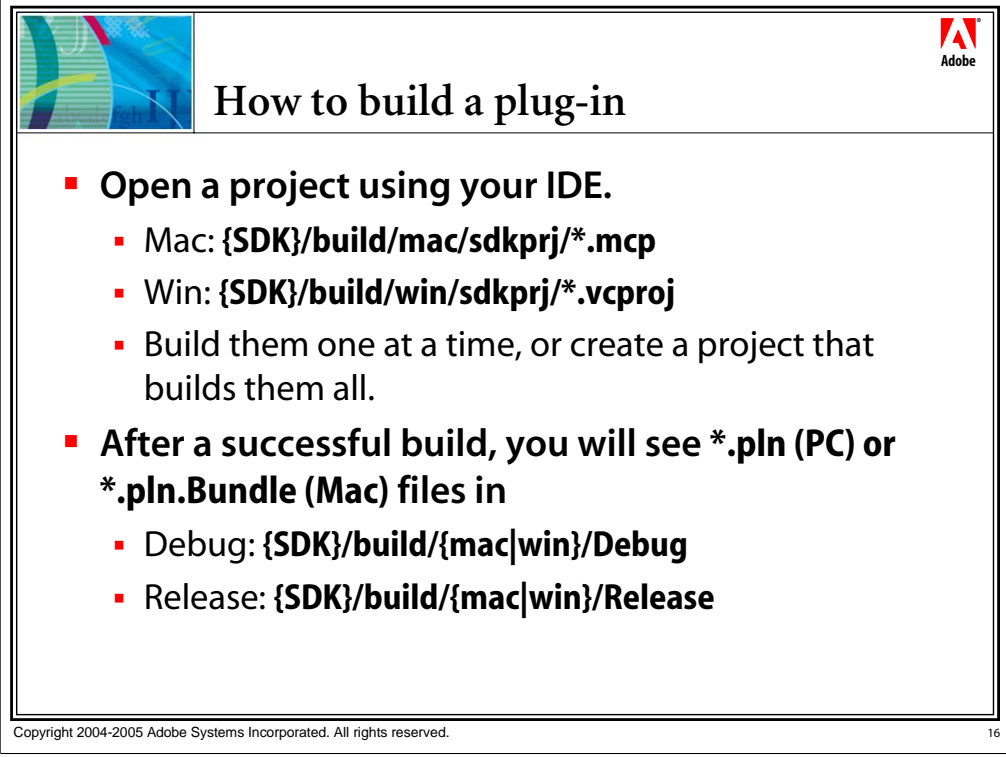


Preparing the IDEs

- **Win: Visual C++ .NET 2003**
 - See *Porting Guide*:
 - “How do I set up Visual C++ .NET 2003 to access tools required to build SDK plug-ins?”
- **Mac: CodeWarrior**
 - See *Porting Guide*:
 - “How do I set up Metrowerks CodeWarrior 9.2 to compile plug-ins?”
 - “How do I set up CodeWarrior to access tools required to build SDK plug-ins?”

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved.

15



How to build a plug-in

- **Open a project using your IDE.**
 - Mac: **{SDK}/build/mac/sdkprj/*.mcp**
 - Win: **{SDK}/build/win/sdkprj/*.vcproj**
 - Build them one at a time, or create a project that builds them all.
- **After a successful build, you will see *.pln (PC) or *.pln.Bundle (Mac) files in**
 - Debug: **{SDK}/build/{mac|win}/Debug**
 - Release: **{SDK}/build/{mac|win}/Release**

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 16

Once you get the IDE setup properly (as per previous slide), building a sample plug-in is as easy as loading a project in your IDE and selecting the Build command!


•VC++ .NET 2003:

- Select the Active Solution Configuration (“Build >> Configuration Manager” menu, then select the “Active Solution Configuration” pulldown) as either Debug or Release
- Select the “Build >> Build {ProjectName}” menu to build.

•CodeWarrior

- Select the build target in the Project window
- Click on the Make button in the Project window, or press Command-M

After the project completes building, you will see a plug-in file (and possibly debug symbols as well), in the folders mentioned on the slide.

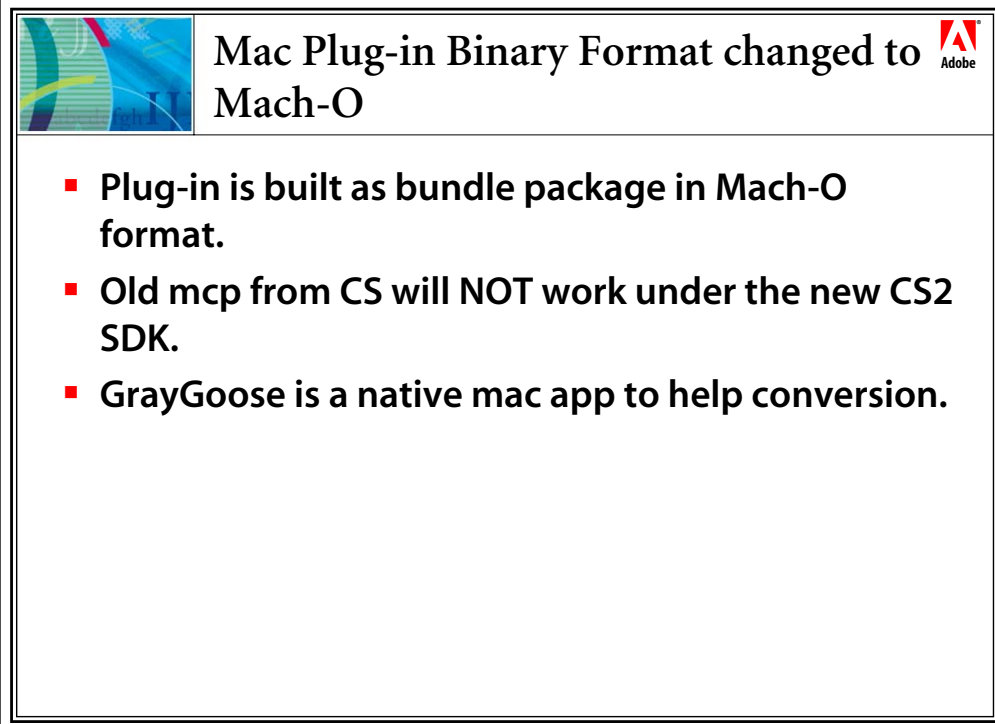


What's New in InDesign CS2/InCopy CS2

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved.

17

The slide features a decorative graphic on the left side consisting of overlapping semi-circles and abstract shapes in shades of blue, green, and yellow. The Adobe logo is positioned in the top right corner of the slide area.





Mac Plug-in Binary Format changed to Mach-O

- Plug-in is built as bundle package in Mach-O format.
- Old mcp from CS will NOT work under the new CS2 SDK.
- GrayGoose is a native mac app to help conversion.

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 18

Note that in CS, plug-in was built as shared library in PEF format.

In the alpha SDK, plug-in was output as framework bundle in Mach-O format, if you've already started your development since alpha SDK, you should be aware that we switched to bundle package now in beta.



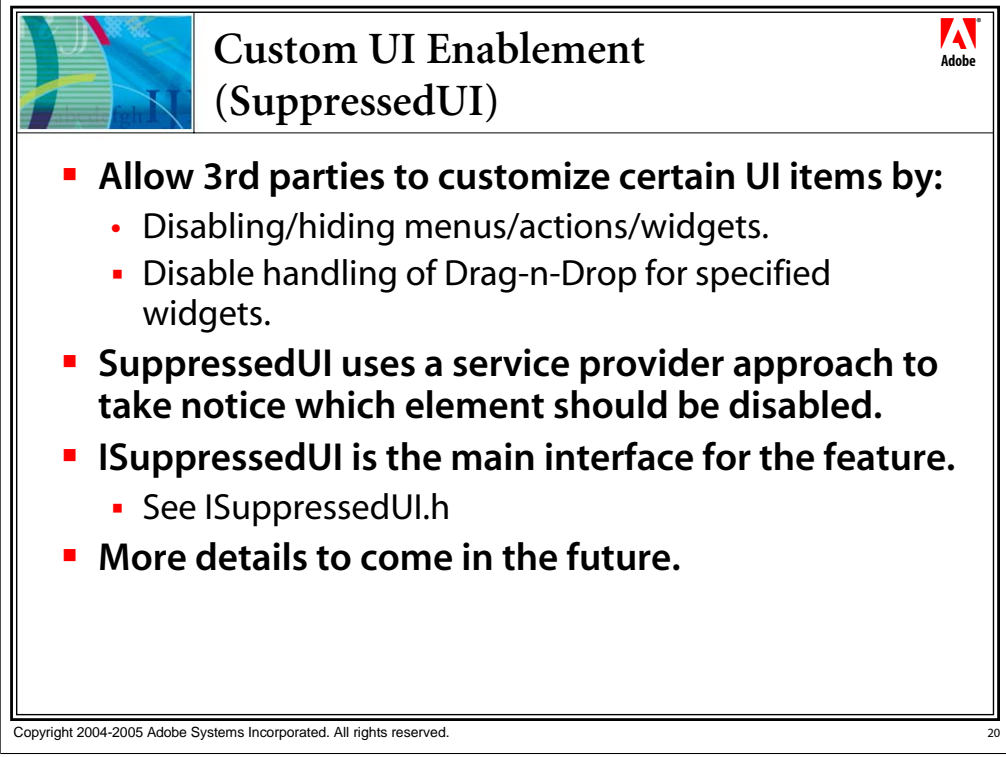
Model/UI Separation

- The entire InDesign code base has gone through thorough model/ui separation process.
- The model code contains things that deal directly with InDesign document model and application workspace, they are separated from the UI so they can be executed without manual work.
- You may need to change your code, but mostly involve compile time issue.
- See *Porting Guide (CS2)* for more details.

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 19

Here are a few guidelines regarding model/ui separation :

1. Interfaces with model and UI methods were refactored to separate interfaces. An example is ILayoutUtils, which had both model and UI methods in InDesign CS, but now it is split into 2 interfaces, ILayoutUtils and ILayoutUIUtils.
2. As a result of some plug-ins being split between model and UI, their associated ID headers are separated according to the nature of the IDs too. For example, the old LayoutID.h now is split into LayoutID.h and LayoutUIID.h. Some of the UI related boss, interface, and implementation IDs, such as kLayoutWindowBoss, are moved to the LayoutUIID.h.
3. Global function QueryPreferences has new signatures.
4. If you are planning on developing plug-ins for InDesign Server CS, you may refer to a technote titled "InDesign Server Plug-in Techniques" for more details on how to separate your plug-ins that mix model and UI components into separate plug-ins.



**Custom UI Enablement
(SuppressedUI)**


- **Allow 3rd parties to customize certain UI items by:**
 - Disabling/hiding menus/actions/widgets.
 - Disable handling of Drag-n-Drop for specified widgets.
- **SuppressedUI uses a service provider approach to take notice which element should be disabled.**
- **ISuppressedUI is the main interface for the feature.**
 - See ISuppressedUI.h
- **More details to come in the future.**

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 20


There are 2 provided ways for suppressing UI elements.

- The first way is to provide an implementation of the ISuppressedUI interface in a service provider boss. There can be several service provider bosses of this type that have ISuppressedUI implementations, each having the chance to suppress every widget, menu, etc. If any of the suppressedUI services respond that they would like to suppress a given UI element, then that UI element will be suppressed.

- The second way is using a default XML based ISuppressedUI implementation that will read in an XML file you provide, this XML will provide information on what UI elements to disable. A default XML based implementation is provided, but you still have to create the service provider boss. In this boss, you will also aggregate an ISysFileData interface that contains the path to the XML file.



Object Styles

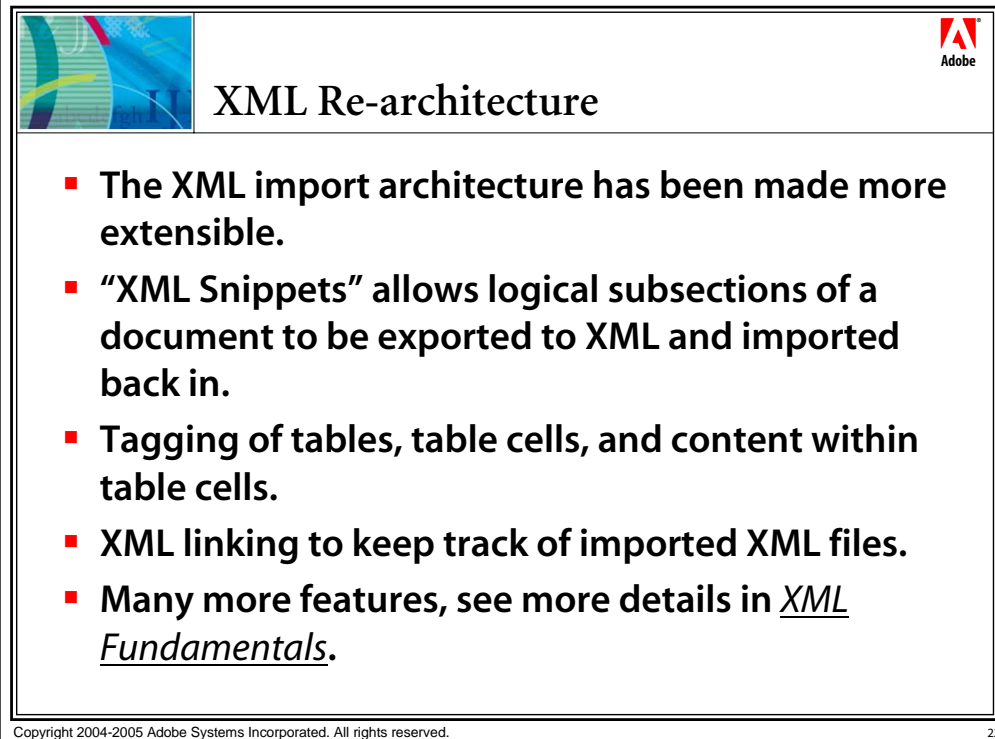


- Collections of object-level appearance properties such as fill/stroke color, transparency, drop shadow.
- Concept similar to paragraph and character styles.
- `IObjectStylesSuite` is the main suite interface on selection boss for most common features.
- `IObjStylesAttrHandler` is used to access the object style target through the model.

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved.

21

Editing a object style updates the objects with that style applied. Style management tasks such as redefining, synchronizing, loading, deleting are supported.



The slide is titled "XML Re-architecture" and features a blue and green abstract graphic on the left and the Adobe logo on the right. The main content is a bulleted list of features. At the bottom, there is a copyright notice and a small page number "22".

XML Re-architecture

- The XML import architecture has been made more extensible.
- “XML Snippets” allows logical subsections of a document to be exported to XML and imported back in.
- Tagging of tables, table cells, and content within table cells.
- XML linking to keep track of imported XML files.
- Many more features, see more details in *XML Fundamentals*.

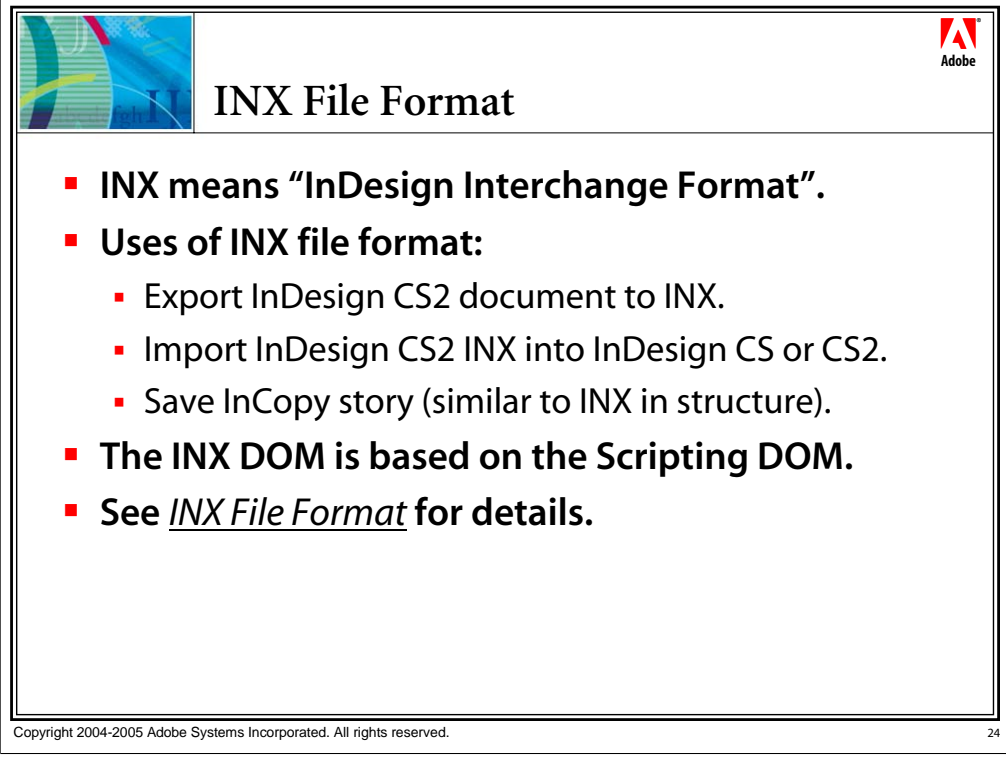
Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 22



Shared Resources

- **Save/Load application preferences to an external file.**
- **ISnippetExport::ExportAppPrefs will export application preferences to a stream.**
- **ISnippetImport::ImportFromStream will import the preferences.**
- **You can specify which elements you want exported.**
- **IAppPrefsImportDelegate, and IAppPrefsExportDelegate provide customization for the import/export.**

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 23



The slide is titled "INX File Format" and features the Adobe logo in the top right corner. It contains a bulleted list of key information about the INX file format. The list includes: INX means "InDesign Interchange Format"; uses of INX file format (exporting CS2 to INX, importing CS2 INX to CS or CS2, and saving InCopy stories); the INX DOM is based on the Scripting DOM; and a reference to the *INX File Format* for details. The slide footer contains the copyright notice "Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved." and the number "24".

INX File Format


- **INX means "InDesign Interchange Format".**
- **Uses of INX file format:**
 - Export InDesign CS2 document to INX.
 - Import InDesign CS2 INX into InDesign CS or CS2.
 - Save InCopy story (similar to INX in structure).
- **The INX DOM is based on the Scripting DOM.**
- **See *INX File Format* for details.**

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 24


Every InDesign document is described by a tree structure of Scripting objects, rooted at a document object. For example, the document will have several children that are color objects, spreads, and so on. Besides having children, each object can have a set of properties. A color will have properties that describe its name, color model, and color values (among other things). When we export an InDesign document to an INX file, we translate the value of every object and property into XML. When we import an INX file, we reverse the process.

The important things to understand about INX are:

1. We create an entirely new InDesign document from a "foreign" file format (the INX). Because of differences between InDesign releases, the resulting document may not look exactly like the original. For example, if the composition algorithm changed, there could be slight differences in how text is laid out in a text frame.
2. The INX file might contain objects or object properties that did not exist in an earlier version of InDesign. These objects and properties will be ignored during import.



INX File Format Family: InCopy .incx file





- InCopy file format (.incx) is based on .inx.
- InCopy CS2 cannot read .inx.
- InCopy CS2 still can read .incd.
- More info to come in the future.

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved.

25

When you save a InCopy story in InCopy CS2, the save dialog defaults to a new file format called “InCopy Interchange”.

Note that the old InCopy CS story file format is still supported.



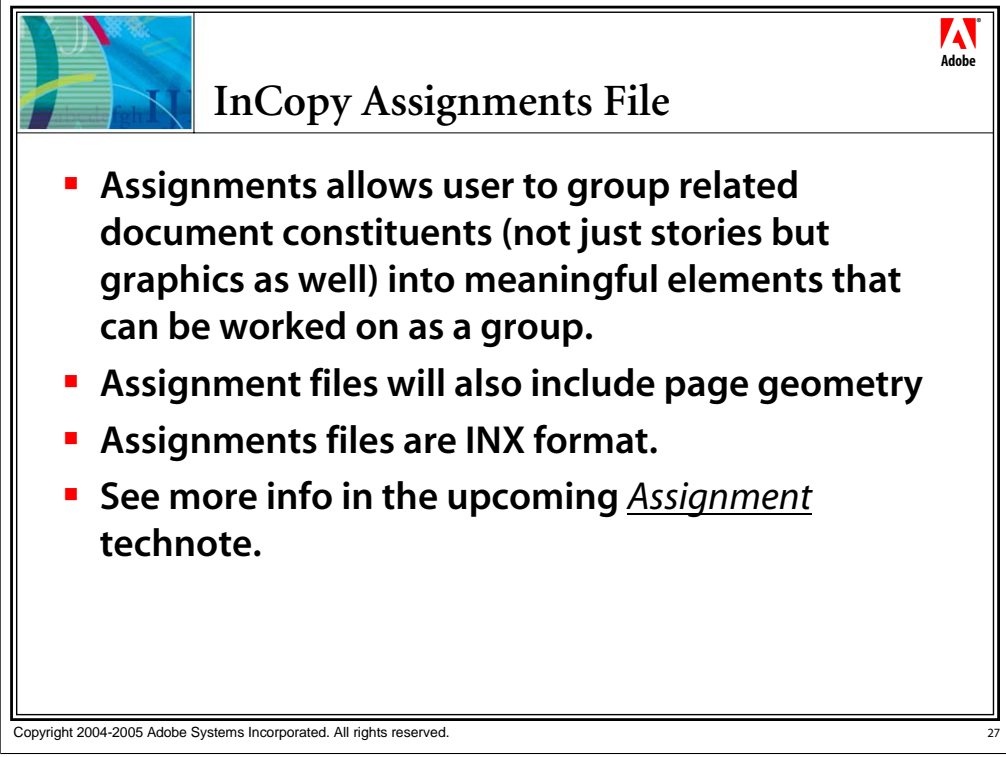
Changes to File Related APIs

- **SysFile has been replaced by the new IDFile class.**
- **AFL (Adobe File Library) is a new library that provides classes for the manipulation of files, directories and paths, with a common API for Mac and Windows.**
- **The FileUtils and FileUtility classes have extensive changes.**
- **See *Using Adobe File Library* for more information.**

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 26

There are a few things to know about the new file library:

1. AFL is a new library that provides classes for the manipulation of files, directories and paths, with a common API for Mac and Windows. In addition, the library has utility classes that support file, path and OS specific operations.
2. SysFile is gone from the code base, replaced by the new IDFile class.
3. The IDFile class subclasses from the AFL's AFile class, and supports file and directory creation, deletion, and the retrieval and setting of system attributes.
4. The FileUtils and FileUtility classes have extensive changes. However, the behavior of the existing methods in both classes has not changed.
5. The rule is never to treat a SysFile or IDFile as a PMString. That assumption was blindly made in some Windows-specific code in the InDesign CS SDK, but has been corrected.



The slide features a decorative graphic on the left with blue and green abstract shapes. The Adobe logo is in the top right corner. The title 'InCopy Assignments File' is centered at the top. Below the title is a bulleted list of four items. At the bottom left, there is a small copyright notice, and at the bottom right, the number '27' is displayed.

InCopy Assignments File


- Assignments allows user to group related document constituents (not just stories but graphics as well) into meaningful elements that can be worked on as a group.
- Assignment files will also include page geometry
- Assignments files are INX format.
- See more info in the upcoming *Assignment* technote.

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 27


Assignment files will include page geometry, so InCopy users can see the content and layout of the frame they are editing, without impacting the time it takes to open an InDesign file.

When exporting a group of items from InDesign, an additional file—the assignment file—will be created that contains links or pointers to the grouped page elements and any transforms on those elements. This lets the user open a single file in InCopy and have editorial access to multiple stories.

Any stories in an Assignment are exported as InCopy files (INX format.) Geometry information and the relationship of the files are held in the Assignment File (also an INX format file.)





Japanese Features Supported in InCopy CS2-J



- **Some Japanese features available in InDesign are now available in InCopy J.**
- **Mainly on the following text features. Layout features are not available.**
 - Kenten, Kinsoku*, Mojikumi*, Ruby, Jikei (Glyph), Shatai, and TateChuYoko.
 - * with only limited feature available. you can only apply Kinsoku/Mojikumi sets that were defined in InDesign.

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 28



These features will only be displayed in the Layout view. Galley/Story views will be similar to the InDesign Story Editor.



Porting Overview

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved.


29



Compiler Settings

- **On PC, your CS vcproj file should work without modifications.**
- **On Mac, project settings need to be modified to accommodate the binary format and plug-in package format changes.**
 - Some old public libraries are gone, and new frameworks need to be added.
 - Should use Mach-O linker now.
 - Plug-in should be built as bundle package.
- **Porting Guide has detailed compiler settings information.**


Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 30




Porting Your Plug-in from CS to CS2

- 1. Backup your old CS version project.**
- 2. Duplicate your project (for Windows, you will use Graygoose to convert a project) and sources in the new SDK.**
- 3. On the Mac, convert the mcp file into the new SDK sdkprj folder using tools such as DollyXs or GrayGoose. Make sure all the project settings are set according to the Porting Guide settings.**
- 4. Build your project, fix any compiler errors.**
- 5. Test your plug-ins in the debug build to see if there are any run time issues.**

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 31




Porting Your Plug-in from CS to CS2: Tools




- **DollyXs is the project generation tool.**
 - It generates CS2 SDK compliant projects and associated template-based source files.
 - Revised in CS2 to include more features.
 - New UI.
 - It allows you to create project files and source where you specify, not necessarily in the SDK folder.

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved.

32



Porting Your Plug-in from CS to CS2: Tools (Cont'd)




- **GrayGoose will convert your old CS mcp into CS2 compatible project.**
 - Supports multiple file batch conversion.
 - Removes/Adds (un)necessary libraries/frameworks to your project(s).
 - Ready to build after a successful conversion.
- **GrayGoose is Mac only.**


Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 33

Since you do not need to convert your Visual Studio .vcproj files from CS to CS2, Snowgoose is not needed and Graygoose is for the Mac only.

There are a few precautions to be aware of when using Graygoose, so make sure you read the help file before you process a project.




Porting Your Plug-in from CS to CS2: Porting Issues




- The changes that took place when porting SDK sample code have been captured in the Porting Guide.
- **When you see compiler errors:**
 - Check out the Troubleshooting section first.
 - Then check out the Porting Recipes.
- **We will take a look at some porting examples in another training session.**

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved.

34



Summary



- **In this session, we have learned:**
 - What's available in the InDesign CS2/InCopy CS2 Combined SDK.
 - Mac project file setting has changed quite a bit.
 - Use the tools provided by SDK to make sure you have correct settings.
 - Some new API areas that you should be aware of.
 - Basic porting strategy.

Copyright 2004-2005 Adobe Systems Incorporated. All rights reserved. 35

