



- Welcome to the session “Programming the InDesign CS2 API for XML”.
- COPYRIGHT NOTICE BELOW.
- ---
- Copyright 2005 Adobe Systems Incorporated. All rights reserved.
- 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.




Pre-requisites

- **Cross-media training session 17 from CS**
 - <http://partners.adobe.com/public/developer/indesign/sdk/>

2005 Adobe Systems Incorporated. All Rights Reserved. 2




- This session expects you to be familiar with the contents of Cross-media training session (number 17) from the InDesign CS SDK training series.
- Visit the partners.adobe.com website to obtain this presentation.




Goals of the training session

- Highlight changes in the XML subsystem brought in for CS2
- Consider changes in XML model
- Introduce new XML import architecture
- Highlight content updates in CS2 SDK relating to XML

2005 Adobe Systems Incorporated. All Rights Reserved. 3




- The main objective of this training session is to highlight changes in the XML subsystem to support the new XML features of InDesign CS2.
- There have been small changes to XML model to support new features, which we consider briefly
- We introduce the XML import architecture of InDesign CS2, which provides several extension points for third parties
- We describe some new and updated content in CS2 SDK, both sample code and documentation, relating to XML




Outline

- XML-related concepts in InDesign CS2
- XML model changes
- XML import architecture
- XML resources in CS2 SDK
- References

2005 Adobe Systems Incorporated. All Rights Reserved. 4




- **To meet the objectives of the training session, the content is structured as follows**
 - Firstly, we consider some of the key XML-related concepts in inDesign CS2.
 - We consider very briefly the XML model for CS2. The XML model refers to how XML-related information is represented in an InDesign document. We discuss some additions to the XML model for CS2 to support the new features of InDesign CS2
 - We introduce the XML import architecture for InDesign CS2, and describe the extension patterns that third parties can implement to participate in the XML import process
 - Export of XML has not changed significantly since InDesign CS so we won't consider that particular function
 - We consider the XML-related resources available in CS2 SDK and we highlight changes in that content since the indesign CS SDK
 - We'll identify references where you can read more about programming the XML subsystem with InDesign CS2 API



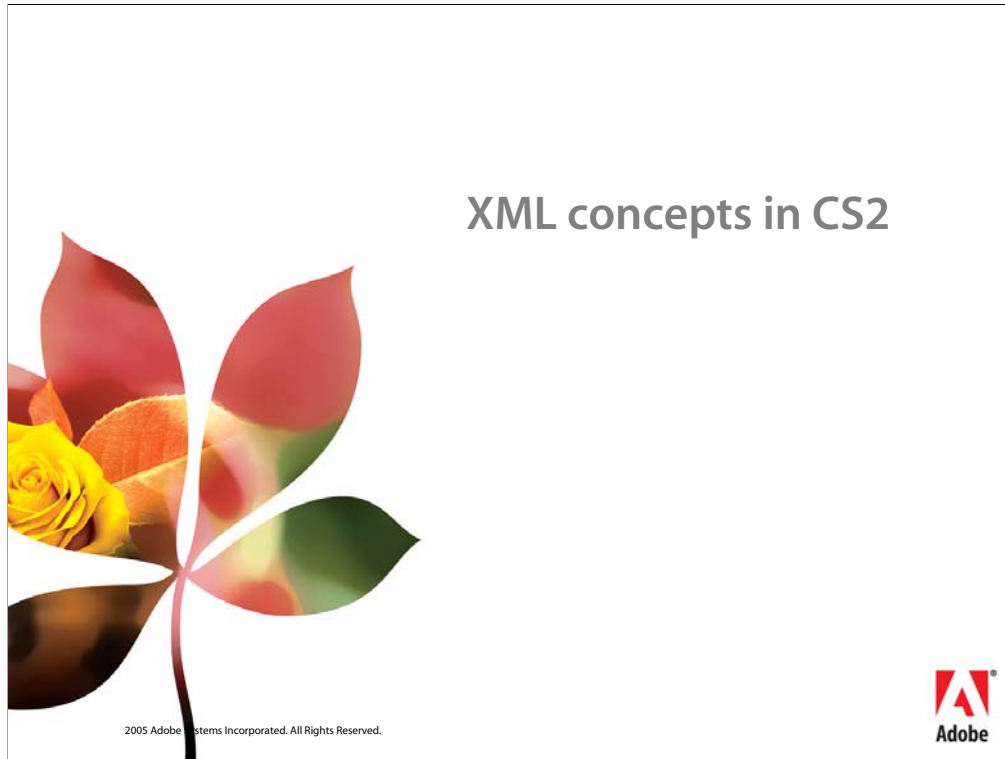
Conventions used


- **File paths**
 - **{SDK}/folder/file.ext**
 - **{SDK}/build/.../sdkprj/**
 - **{SDK}/build/(mac|win)/sdkprj/**
- **URLs**
 - <http://www.adobe.com/>
- **Document titles**
 - *Programming Guide*

2005 Adobe Systems Incorporated. All Rights Reserved. 5



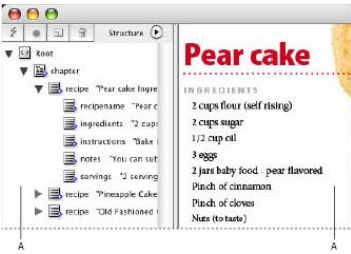
- **Conventions used in the slides.**
- **File paths are shown in a bold condensed font.**
 - SDK shown in curly brackets to signify the folder where you installed the InDesign Products CS2 SDK
 - Ellipsis to indicate arbitrary component of path
 - (a|b|...) to indicate alternate choices for a component
- **URLs are in blue and underlined.**
- **Document titles are shown underlined and italicized.**





Logical structure


- XML info in an InDesign CS2 document
- Captures user-defined relationships
- Importing XML modifies logical structure
 - Tagging changes/ adds logical structure
 - XML export based on logical structure




A (left): structure view, rendering of logical structure

A (right): layout view, showing placed content

2005 Adobe Systems Incorporated. All Rights Reserved. 7

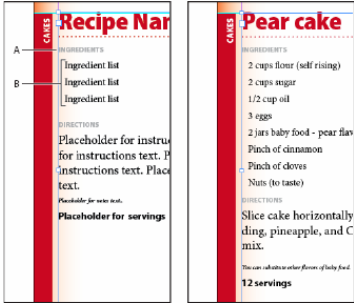


- Logical structure is the term we use for the XML-related information stored in an InDesign CS2 document. Logical structure is composed elements, the DTD declaration (if any), XML comments, processing instructions etc, and represents relationships between them
- One key point is it captures user-defined relationships and the semantics of user data; for instance, articles, headline, byline, body copy etc.
- End-result of XML import is (typically) modified logical structure
- You can also change the logical structure by tagging content items, manually or programmatically, or creating new elements as container elements in the logical structure
- One of key points about logical structure is that it forms the basis of XML export, which is based on *only* the logical structure



XML templates

- InDesign doc. with tagged page items/ text
- Define patterns to match against the input
- Placeholders for text and graphics




Recipe Nar

Pear cake


A: static text, line breaks and so on

B: tagged placeholder text, replaced on import

2005 Adobe Systems Incorporated. All Rights Reserved. 8




- An XML template is an InDesign document with tagged placeholders and placeholder text into which XML can be imported. Key point is that it defines a pattern to match against incoming XML.
- Graphic frames can be tagged as placeholders for images. Stories and text ranged can be tagged as placeholders for textual content, and now tables and table cells can be tagged as placeholders for incoming XML-based content.
- With InDesign CS2, it is now possible to customize how matching against the XML template occurs, and there are some interesting possibilities for third parties in this area.




XML features new to InDesign CS2

- **Filtering and matching options**
 - Discard incoming XML unmatched in template
 - Discard template elements unmatched in incoming XML
 - Import repeating elements
 - See Adobe InDesign CS2 Help
- **Tagging of tables/ cells**
- **Tagging of in-line graphics**
- **Option to create datalink to XML file imported**

2005 Adobe Systems Incorporated. All Rights Reserved.




- There are some new features in InDesign CS2 that enable incoming XML elements to be filtered, and provide new kinds of matching of the incoming XML against an XML template.
- See InDesign CS2 for more help
- Discard incoming XML elements not matched in the XML template
- Discard XML template elements not matched in incoming XML
- (Options)
- Import repeating elements
- Tagging tables
 - Table and individual cells can be tagged
- In-line graphics could not be tagged in InDesign CS
- XML linking
 - Can create link to XML files on import



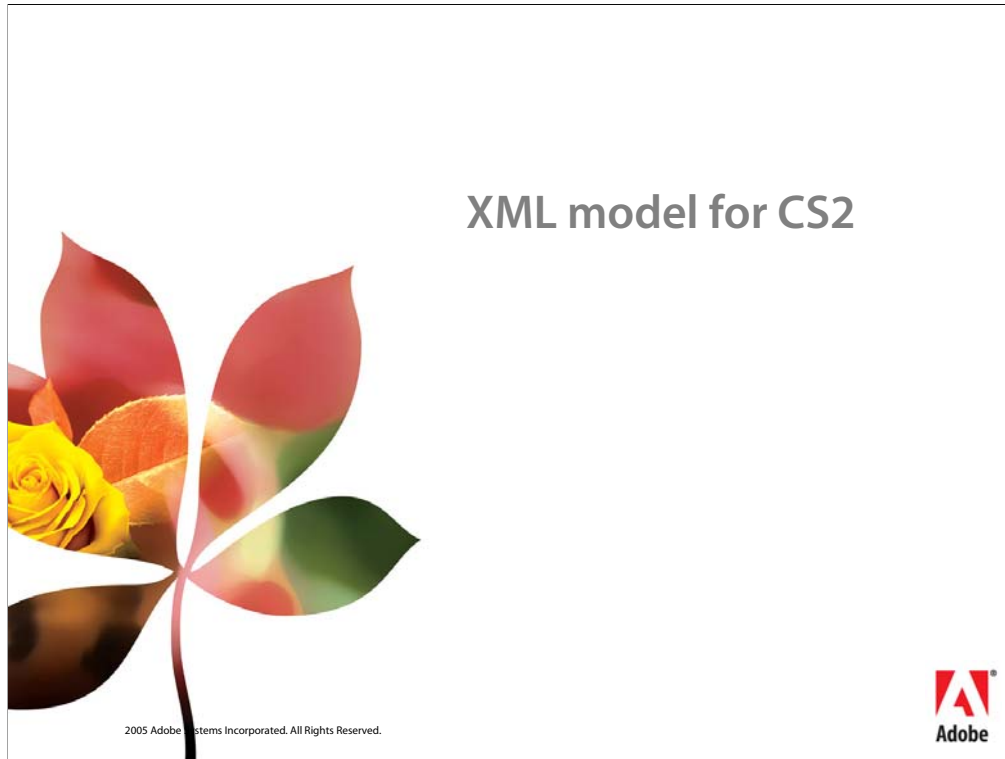
CS2 support for new features

- **XML model**
 - Enhanced to support the new features
 - Small changes in client API
- **XML import process**
 - Enhanced to support the new features
 - Small changes in client API relating to import
 - New hooks for third parties on XML import


2005 Adobe Systems Incorporated. All Rights Reserved. 10



- To enable new features to be added to InDesign CS2, some enhancements to the existing API of InDesign CS were required. The CS API has been extended rather than revised, so you should find that there are few if any changes required to client code.
- We can partition the enhancements that affect third parties into:
 - Those that affect the XML model; how XML information is represented in a doc
 - Those that affect how it is processed on import.
- The model API has not changed, but some additional capabilities have been added, to support tagging of new content items like tables.
- The XML import architecture has been improved from a third party perspective.
- In InDesign CS2, there are now several new hooks for third-parties on XML import; that is, places where you can take control as a third party during XML import, if you implement the appropriate extension pattern.
- In addition, there are new APIs to store XML-import related preferences and create UI for these preferences, which attempt to simplify and abstract how preferences are stored and manipulated.




- "XML Model for CS2"




InDesign XML model

- **How XML info stored in an InDesign CS2 document**
- **XML element represented by IIDXMLElement**
 - Represents node in the logical structure
 - All nodes in logical structure have interface IIDXMLElement
 - Can traverse logical structure via this interface
- **Change logical structure via facades/suites**

2005 Adobe Systems Incorporated. All Rights Reserved. 12




- What we mean here by the XML model is how XML information is represented in an InDesign CS2 document; we already introduced logical structure, the elements and relationships between them. The key part of the XML model are XML elements, and relationships that these elements have with content items. There are other objects such as tag-style maps and tags that we also consider part of the XML model.
- By default, each InDesign CS2 document has logical structure consisting of a single element with the default tag-name of Root. This element has no content and is not associated with anything in the layout.
- The key abstraction in the InDesign API for XML is the interface IIDXMLElement, exposed on boss objects that represent the logical structure of an InDesign CS2 document.
- You can traverse the logical structure of an InDesign CS2 document using IIDXMLElement; it is somewhat more straightforward to traverse the logical structure, using this single interface, than to traverse the layout model, say traversing the spreads and content in a document.
- You manipulate the XML model via command facades/ suites; the XML commands are named IXML<whatever>Commands, the suites IXML<whatever>Suite
- AT THIS POINT LOOK AT API DOCS
- AT THIS POINT LOOK AT INDESIGN CS2



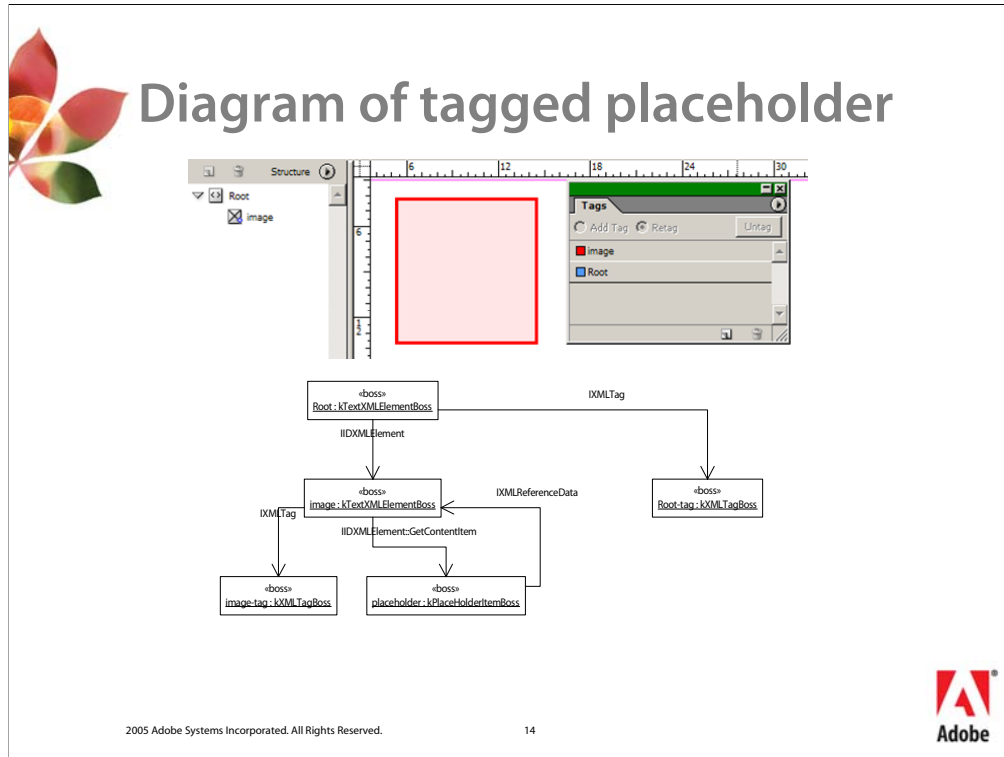
XML-related associations

- **Elements associate with tags (kXMLTagBoss)**
 - See IXMLTag
- **Elements (IIDXMLElement) associate with content items**
 - XMLContentReference stores information about association, if any
- **Content items associate with elements**
 - XMLReference stores information about associated element, if any
 - No universal marker for a content item, IXMLReferenceData is closest
 - But see IXMLUtils::IsTaggablePageItem


2005 Adobe Systems Incorporated. All Rights Reserved. 13



- One key aspect of the XML model are the associations elements can have with other objects.
- XML elements (IIDXMLElement) associate with tags (kXMLTagBoss/ IXMLTag).
- For instance, suppose that there are several elements with the same tag-name (headline, say); each XML element would point to a single tag, a boss object of type kXMLTagBoss, responsible for storing the string "headline" and a colour in which it would be displayed in the UI.
- Elements can be associated with content items in the layout, if they are placed. The class XMLContentReference represents information about the association from an XML element to a content item in the layout.
- Content items can refer to XML elements, if they are tagged content items.
- The class XMLReference is one way to refer to an XML element. It is somewhat like an UIDRef for XML element
 - You may be able to instantiate the interface IIDXMLElement from XMLReference
- Content items include placeholder graphic frames, images, text ranges, tables, inline graphics and so on. These have quite diverse representations in the model, and there is no simple hierarchy in which all the content items participate. It is difficult to decide what makes something a content item.




- PLACED EPS (WORKAROUND FOR VISIO). NOTE THE PREVIEW IN POWERPOINT IS POOR BUT THE PDF PRINTS FINE.
- This object diagram shows instances of associations between classes involved in representing the XML structure for a tagged placeholder for an image.
- You can see examples of the associations mentioned in the previous slide:
 - For instance, the `<Root>` element, instance of `kTextXMLElementBoss`, maintains a reference to it's child(ren) via `IIDXMLElement`
 - The `<Root>` and `<image>` elements have links to tags (`IXMLTag`). The tag stores the element name, and also how it appears in the UI (see `kXMLTagBoss` in API docs)
 - The `<image>` element maintains an association (via `IIDXMLElement`) with a content item in the layout (`kPlaceholderItemBoss`).
 - The placeholder in the layout (via `IXMLReference`) has an association instance connecting it to the element representing its markup



XML model API in CS2


- **Support for tagging tables and cells**
 - XMLContentReference added
 - Generalises references to non-UID based objects
 - Removes need for IIDXMLElement::GetContentItem
- **Changes additive, client code mostly unaffected**
 - Move to using XMLContentReference going forward

2005 Adobe Systems Incorporated. All Rights Reserved. 15



- In InDesign CS, content items had to be UID-based, with the notable exception of text ranges.
- To support tagging of table cells, some mechanism to support references from XML elements to non-UID based content items was introduced. The abstraction XMLContentReference supports a reference from an element (IIDXMLElement) to a content item, which can be UID-based, as in InDesign CS, or a non-UID object like a table cell.
- The changes are additive, meaning that existing client code would not break, but you should use IIDXMLElement::GetContentReference and XMLContentReference in preference to GetContentItem







Importing XML

- **Import XML command**
 - kImportXMLFileCmdBoss has IPMUnknownData interface to specify import source
 - Was IImportFileCmdData in CS
- **New boss class to carry XML import settings**
 - kImportXMLDataBoss
- **See API docs for kImportXMLFileCmdBoss**

2005 Adobe Systems Incorporated. All Rights Reserved. 17




- There are some small changes in the client API for import of XML compared to CS that will break existing client code if you already use XML import.
- There has been a small change to the import XML cmd to support the new import mechanism in InDesign CS2
- In InDesign CS API, you would have used IImportFileCmdData to specify an import source
- In InDesign CS2 API, you create an instance of kImportXMLDataBoss, and parameterize it to specify the import source, then pass a reference to this to the import XML cmd. This is well documented in the API documentation; see docs for kImportXMLFileCmdBoss
- **AT THIS POINT SWITCH TO THE API DOCUMENTATION AND SHOW PAGE FOR kImportXMLFileCmdBoss and follow up couple of links e.g. to kImportXMLDataBoss (specifies import source) and kXMLImporterBoss (governor for import process)**




XML import re-architecture

- **Nature of change**
 - Import now extensible
 - XML import factored into distinct phases
 - Each phase has associated extension pattern
 - Preferences control behaviour at service level
- **Benefits for third-parties**
 - Participate in import via defined extension patterns
 - Clients extend XML import UI without writing ODFRez

2005 Adobe Systems Incorporated. All Rights Reserved. 18




- The XML import architecture for InDesign CS2 was designed to meet the Adobe requirement of adding new features more easily than was possible in CS; it also met a second goal, of improving third-party access to the XML import process.
- XML import now more extensible in CS2 compared to InDesign CS
- The XML import architecture for InDesign CS2 was defined as a set of steps or phases, and extension patterns defined that would come into play at each step. These were based on public interfaces, meaning that if you followed the correct pattern, you could participate at each step as third party.
- Since the process was being broken down into a set of steps, it was desirable for the processes involved in each step to be able to maintain their preferences individually, to allow them, for instance, to be turned on and off independently.
- Also, some straightforward and consistent way to have the preferences accessible through the UI was identified as a requirement. Robust support for dealing with documents containing data from missing plug-ins was another requirement.



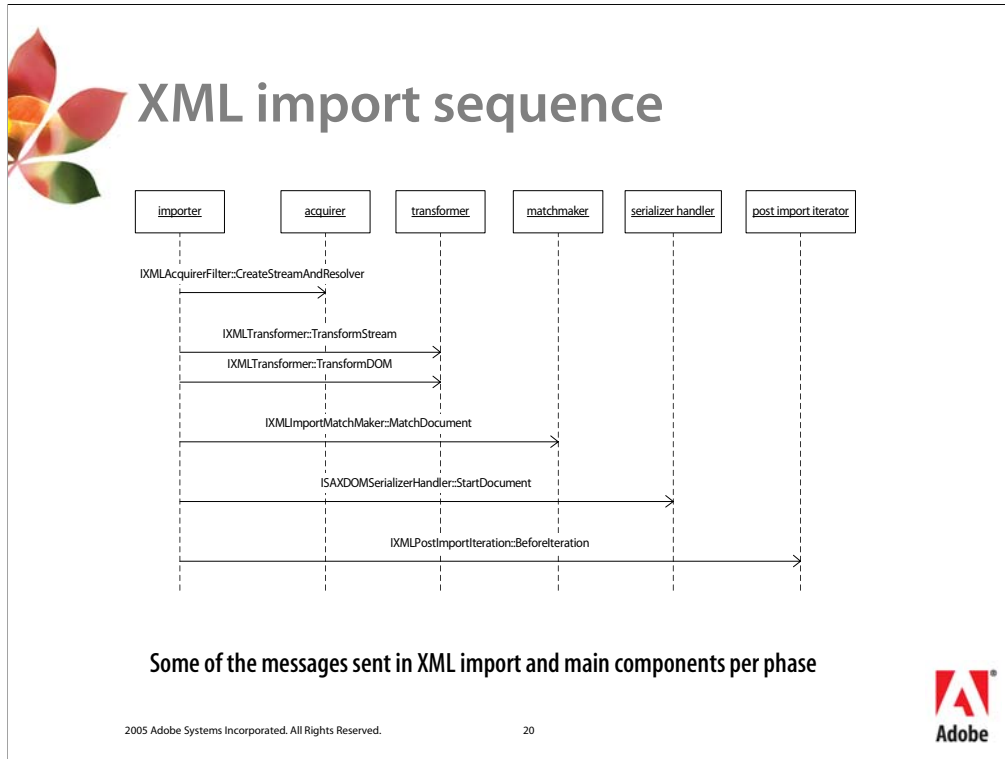
XML import phases

- **Distinct phases during XML import in CS2**
 - XML acquisition
 - XSLT/ tree manipulation
 - Element matching
 - Content processing
 - Post import processing
- **Each phase has an extension pattern**


2005 Adobe Systems Incorporated. All Rights Reserved.



- There are now distinct phases during XML import. Each phase is an opportunity for a third party to take control of the import process.
- We consider these phases in more detail in the slides following this one.
- Briefly, the first stage, XML acquisition, allows you to specify where the XML data comes from. The next stage, XSLT/ tree manipulation, lets you modify the incoming XML before it comes into the document. The next stage, element matching, lets you control how incoming XML is matched against the elements in the XML template. The next stage, content processing, is where content is actually added to (technically, serialized into) the document. The final stage, post-import processing, occurs after the content has been added to the document.
- Each phase is associated with a well-defined extension pattern.
 - Let you take control at well-defined points during XML import, if you implement them
- The extension patterns are implemented as services (IK2ServiceProvider). The import architecture allows for multiple instances of each service type to contribute at each phase of import.




- PLACED EPS (WORKAROUND FOR VISIO). NOTE THE PREVIEW IN POWERPOINT IS POOR BUT THE PDF PRINTS FINE.
- This slide shows the XML import sequence as a UML sequence diagram. Note how messages are sent out by the importer, and instance of kXMLImporterBoss, to the components that participate in XML import.
- Like all models, there is some degree of simplification here.
- Firstly, not all the messages have been shown
- Secondly; as mentioned previously, could be several instances of a particular component involved, but the sequence is drawn as if there is one and only one instance of each component. There could be several post-import iterators, for instance, all involved in doing something like clean-up of the logical structure after import.




XML acquisition phase

- **Use this phase to get or create an XML stream**
 - Existing acquirers: open stream onto file
- **Existing use**
 - Open stream onto XML file
- **Extension pattern is XML acquirer-filter**
 - Provided interface `IXMLAcquirerFilter`
 - ServiceID is `kAcquireXMLService`

2005 Adobe Systems Incorporated. All Rights Reserved. 21




- **The first phase we mentioned is the XML acquisition phase.**
 - Also have opportunity to pass back an entity resolver (`ISAXEntityResolver`)
- **CURRENT USE:** We currently use this phase to open a text file to get a stream and return this stream for the import process.
- **A POTENTIAL USE for Third parties** could use this to read information from a database into a memory buffer (say) and create a stream onto this memory buffer, and then import XML from this memory-based stream.
- This could allow users to link directly to a database rather than having to create an interim XML file as they do today.
- Provided (service) interface is `IXMLAcquirerFilter`
- Required interface `IK2ServiceProvider` with `SERVICEID` of `kAcquireXMLService`
-




XSLT/ tree manipulation phase

- **Use this phase to transform incoming XML**
 - Apply XSLT transform
- **Existing use**
 - SDK sample: XDocBookWorkflow
- **Extension pattern is XML transformer**
 - Provided interface IXMLTransformer
 - Can operate on stream or fully constructed DOM
 - ServiceID is kXMLImporterTransformerService

2005 Adobe Systems Incorporated. All Rights Reserved. 22




- **CURRENT USE:** Currently an SDK only feature.
- **POTENTIAL USE:** Clients could use this to process specific XML transformations during the import process. Could be used, for instance, when the records from a database don't map exactly to the tag names in the document.
- Find example in SDK (XDocBookWorkflow) of this being implemented
- Provided interface is IXMLTransformer
- Required interface IK2ServiceProvider with SERVICEID of kXMLImporterTransformerService




Element matching phase

- **Match incoming elements to existing elements in XML template**
- **Existing use(s)**
 - Ignore unmatched incoming/ existing
 - Import repeating elements
 - “Clone repeating text elements” feature
- **Extension pattern is import match-maker**
 - Provided interface `IXMLImportMatchMaker`
 - ServiceID is `kXMLImportMatchMakerSignalService`

2005 Adobe Systems Incorporated. All Rights Reserved. 23




- This phase is when clients match incoming elements to existing elements in the XML template.
- **CURRENT USES:**
 - Ignore Unmatched Incoming (elements in incoming XML)
 - Ignore unmatched existing (in XML template)
 - Importing repeating elements
- Provided interface is `IXMLImportMatchMaker`
- Required interface `IK2ServiceProvider` with SERVICEID of `kXMLImportMatchMakerSignalService`



Content processing phase

- **Put content into elements that were matched during the matching phase**
- **Existing use(s)**
 - Adds text from incoming XML to the document
 - Adds tables to the document
- **Extension pattern is DOM serializer handler**
 - Provided interface: ISAXDOMSerializerHandler
 - ServiceID is kXMLContentHandlerService


2005 Adobe Systems Incorporated. All Rights Reserved. 24



- **CURRENT USES:** Adds text from incoming XML to the document

In the XDocBookWorkflow sample plug-in, an InDesign table is created based on a table in CALS format in the incoming XML.


- **Provided interface is ISAXDOMSerializerHandler**
- **Required interface IK2ServiceProvider with SERVICEID of kXMLContentHandlerService**



Post import processing phase

- **Post-import clean-up stage**
 - Perform operations that could not be done in content handling phase
- **Existing use(s)**
 - Tag to style mapping
 - Attribute to style mapping
- **Extension pattern is post-import iterator**
 - Provided interface: IXMLPostImportIteration
 - ServiceID is kXMLImporterPostImportService


2005 Adobe Systems Incorporated. All Rights Reserved. 25



- **CURRENT USES:**
- Tag to style mapping
- Attribute to style mapping

- **POTENTIAL USES:** Rules handling (similar to email rules) could be done here.
- Object styles could be applied via attributes.


- Provided interface is IXMLPostImportIteration
- Required interface IK2ServiceProvider with SERVICEID of kXMLImporterPostImportService




XML import preferences overview

- **“Global” import preferences**
 - Some from CS (IXMLImportOptions)
 - Some new to CS2 (IGeneralXMLPreferences)
- **Service-level prefs (IXMLImportPreferences)**
- **XML import options pool (IXMLImportOptionsPool)**
 - Accessible by all XML import services
 - Store and retrieve data from these services
 - Handles missing plug-ins gracefully

2005 Adobe Systems Incorporated. All Rights Reserved. 26




- There are XML-related preferences that are what we could call “global”, such as IXMLImportOptions. These preferences could affect any part of the XML import process. There are also service-local preferences.
- Individual services in InDesign CS2 now have their own interface to store their settings (IXMLImportPreferences).
- IXMLImportPreferences aggregated on a service boss class, but the preference data is really stored in an import options pool as part of an opaque chunk of data.
- IXMLImportOptionsPool stored in workspaces (kDocWorkspaceBoss, kWorkspaceBOss)
 - The XML import options pool handles missing plug-in data gracefully
 - When the plug-in that stored the data is not available, nothing bad happens




XML import options user interface

- **Design goals**
 - Have a single prefs dialog for all XML-related services
 - Let implementors add a UI for their pref without ODFRez resources
 - Let implementors access and control other widgets on import options dialog

2005 Adobe Systems Incorporated. All Rights Reserved. 27




- Rather than requiring each distinct XML service to implement its own preferences dialog, for instance, by adding in a panel to the global preferences selectable dialog, it was reckoned that it was worth providing a simple mechanism to support adding your preferences to a single XML import options dialog, ideally without having to define new ODFRez resources.
- You should also be able to control the state of widgets on this import options dialog.



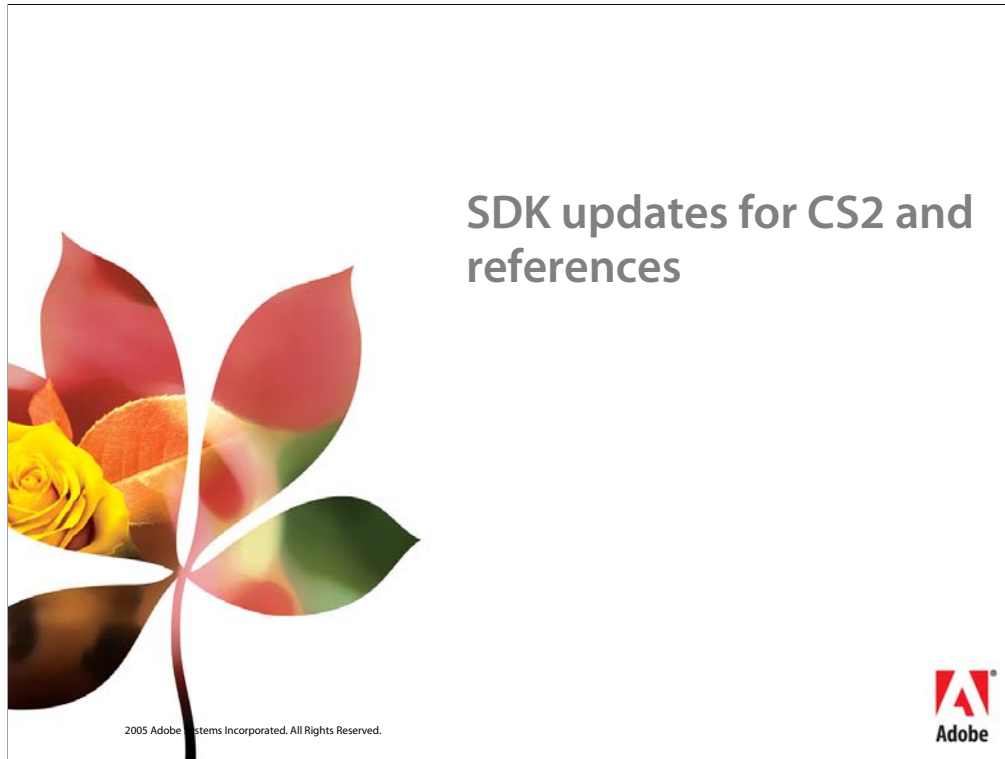
XML import options UI continued


- **Extension pattern is UI import options service**
 - Provided interface is IXMLImportOptionsService
 - Required interface IK2ServiceProvider with ServiceID of kXMLImportUIService
- **Implementation hints**
 1. Re-use kXMLImportUIServiceProviderImpl
 2. Subclass CXMLImportOptionsServiceProvider
 3. Override the Initialize() method
 - Call CXMLImportOptionsServiceProvider::Add...() to add your option (can be a checkbox, radio group or popup) and set its default state
 4. Override the WarnOfChange() method
 - Use this to change your preference value or disable your widget based on changes the user made.
 5. Override the CommitChanges() method
 - Use this to get the final value and set your preference.

2005 Adobe Systems Incorporated. All Rights Reserved. 28



- 1 Define a service boss class with interface IK2ServiceProvider, with ServiceID of kXMLImportUIService. You can re-use kXMLImportUIServiceProviderImpl to do this
- 2 You need to provide an implementation of IXMLImportOptionsService. To do this you can subclass the partial implementation class named CXMLImportOptionsServiceProvider
- 3...
- Not yet demonstrated in sample, but required classes are in public API
- This is described in detail here because currently it is not documented anywhere else, though this should change in an SDK update.







XML documentation

- **Product documentation**
 - Adobe Help > InDesign CS2 > Help Topics > XML
 - <http://www.adobe.com/products/indesign/crossmedia.html>
- **Programming Guide**
 - XML fundamentals (chapter)
 - `{SDK}/docs/guides/`
- **Working with XML**
 - Implementing XML-related use-cases
 - `{SDK}/docs/guides/ww_xml.pdf`
- **API documentation**

2005 Adobe Systems Incorporated. All Rights Reserved. 30




- **XML fundamentals** rewritten
 - Added to *Programming Guide*
- **Working with XML** written to support using the XML-related API
- **API documentation for XML-related API enhanced**
 - XML interfaces largely completely documented
 - Some boss classes well-documented
 - See `kImportXMLFileCmdBoss`, `kXMLImporterBoss`
 - `{SDK}/docs/references/`




XML-related sample code

- **XDocBookWorkflow/UI**
 - XML transformer
 - DOM serializer handler
 - Post-import iterator
 - XML import prefs
- **Finding the code**
 - **{SDK}/build/.../sdkprj/XDocBookWorkflow...**
 - **{SDK}/source/sdksamples/xdocbookworkflow**
 - **{SDK}/source/sdksamples/xdocbookworkflowui**

2005 Adobe Systems Incorporated. All Rights Reserved.



- XDocBookWorkflow shows one way to deal with an XML vocabulary that uses a content model a little different to the one that InDesign CS2 supports out of the box. For instance, it shows one way to import tables defined in OASIS CALS table model format, by using a serializer handler to create Indesign tables on import of the XML.
- It also shows typical uses of a post-import




Sample plug-ins (continued)


- **XMLCatalogHandler implements:**
 - Acquirer filter
 - Entity resolver
 - SAX content handler

- **Finding the code**
 - **{SDK}/source/sdksamples/xmlcataloghandler**

2005 Adobe Systems Incorporated. All Rights Reserved.




- XMLCatalogHandler shows one way to load XML from a file but without importing the XML into the logical structure of an InDesign CS2 document, and shows implementing several extension patterns relating to XML




Sample code snippets

- **SnInspectSelectionXMLProperties**
 - Inspect XML-related properties carried in different selection formats
- **SnImportExportXML**
 - Working with XML import and export as a client
- **SnManipulateXMLElements**
 - Creating and modifying different kinds of elements/ tagged content
- **SnManipulateXMLTags**
 - Working with tags and tag-style mappings
- **SnManipulateStructureView**
 - Open/ close, setting a node selected
- **SnManipulateXMLSelection**
 - Using XML APIs that depend on a selection existing in the structure-view
- **Finding them:**
 - `{SDK}/source/sdksamples/codesnippets`

2005 Adobe Systems Incorporated. All Rights Reserved. 33




- There are several code snippets provided in the CS2 SDK. These cover the same ground as the snippets in the CS SDK and go further in some cases, to explore working with the new APIs.
- A good place to start is with the snippet to inspect the XML related information in a selection.
- These snippets are located in the SDK folder `source/sdksamples/codesnippets`, and can be built through the SnippetRunner project, in `build/(win|mac)/sdkprj`



XML vocabularies

- **Organisations**
 - <http://www.ifra.org> IFRA
 - <http://www.naa.org/> Newspaper Assoc. of America
- **News content**
 - <http://www.newsml.org> IPTC NewsML
 - <http://www.nitf.org> IPTC NITF
- **Advertisements**
 - <http://www.admsl.org> AdsML (IFRA/ NAA)
 - <http://www.adconnexion.org> IFRA adConnexion
- **Metadata**
 - <http://www.adobe.com/products/xmp> Adobe XMP

2005 Adobe Systems Incorporated. All Rights Reserved. 34



- References to some existing XML vocabularies of relevance to publishing are shown.
- NewsML is an XML container format for exchanging news items, representing metadata and providing a protocol for interchanging news items. It does not describe the content itself. For textual news items, you might use NITF format to represent the articles.
- The relationship between AdsML and adConnexion is somewhat similar to that between NewsML and NITF.
- AdsML is an XML container format for advertising transactions, which can describe:
 - The content and metadata of such transactions
 - Protocols (e.g. relating to security) required to support the transactions.
- An AdsML “envelope” can carry both XML as well as legacy formats such as comma separated values (CSV) text files and binary files.



