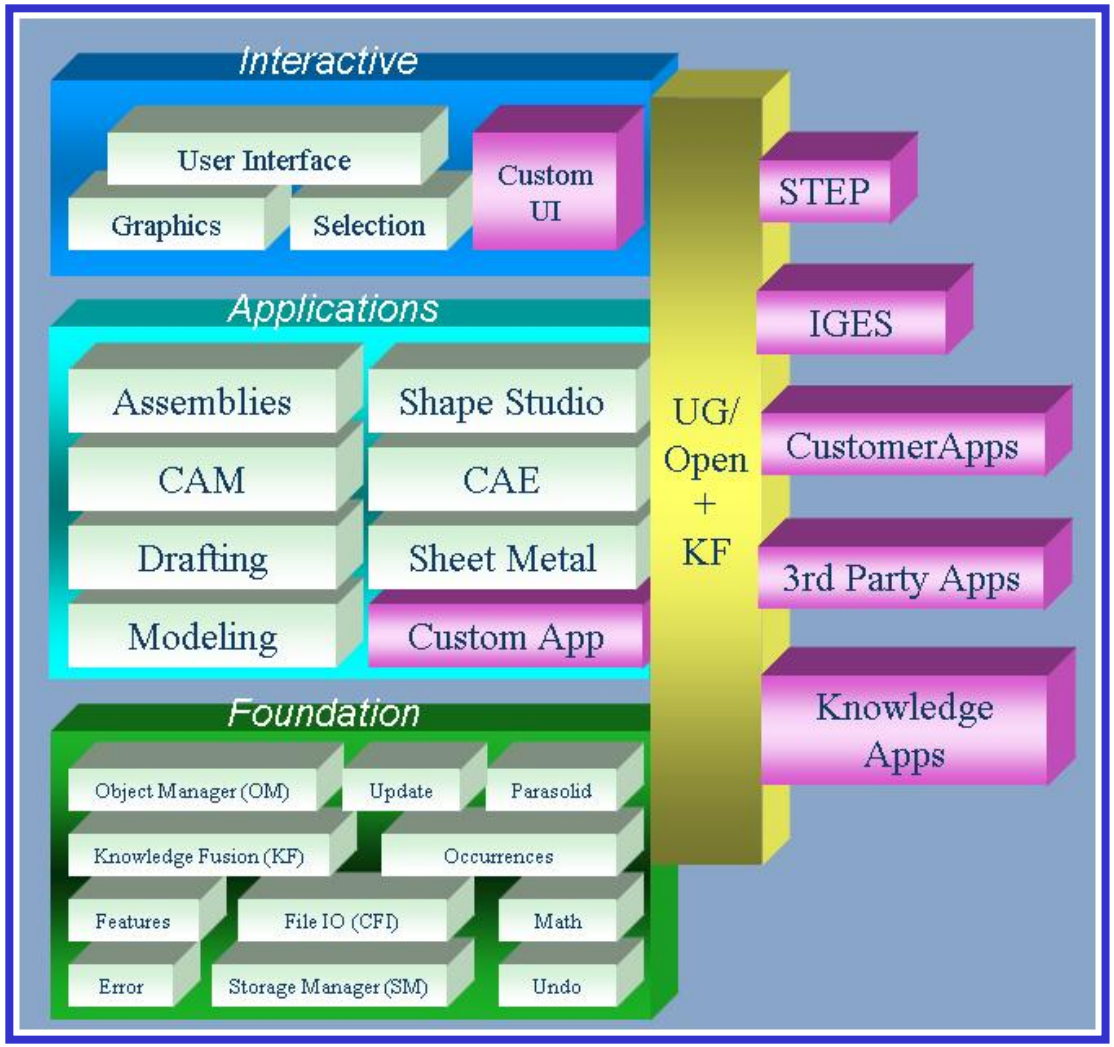
NX Open …*new & cool…*

Rick Rodgers
UGS
rick.rodgers@ugs.com
(314) 264-8866

Topics

- Vision Review *…where we are going…*
- Common API *…more languages, greater coverage…*
- Journaling *…UI improvement, greater coverage…*
- Runtime License Control *…totally new…*
- How to Move Forward *…pulling it all together…*
  - learning the common API
  - turning journals into applications
  - using your old Open C programs
  - migrating from Open I-deas

Vision
Legacy UG Architecture

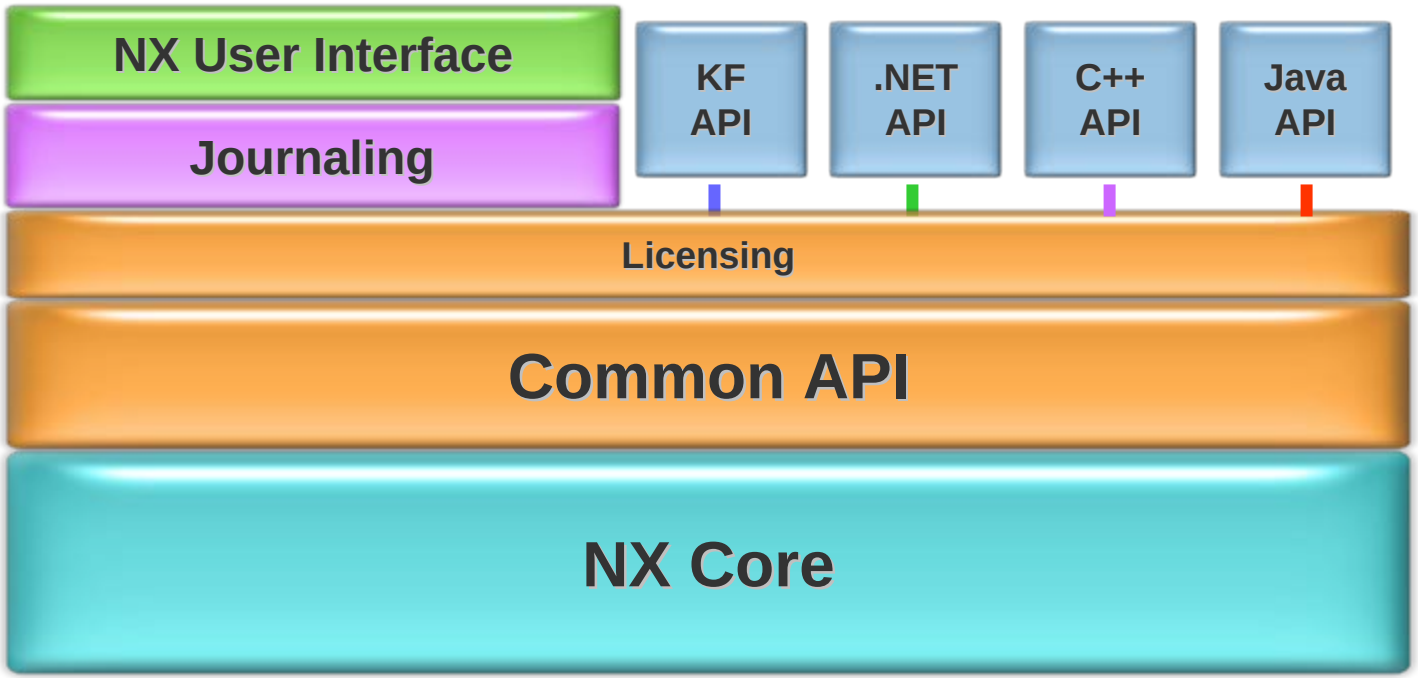


- Expensive for UGS to maintain coverage
- Behaviors not 1-to-1 with UI
- Limited Languages

Still very good & capable APIs…

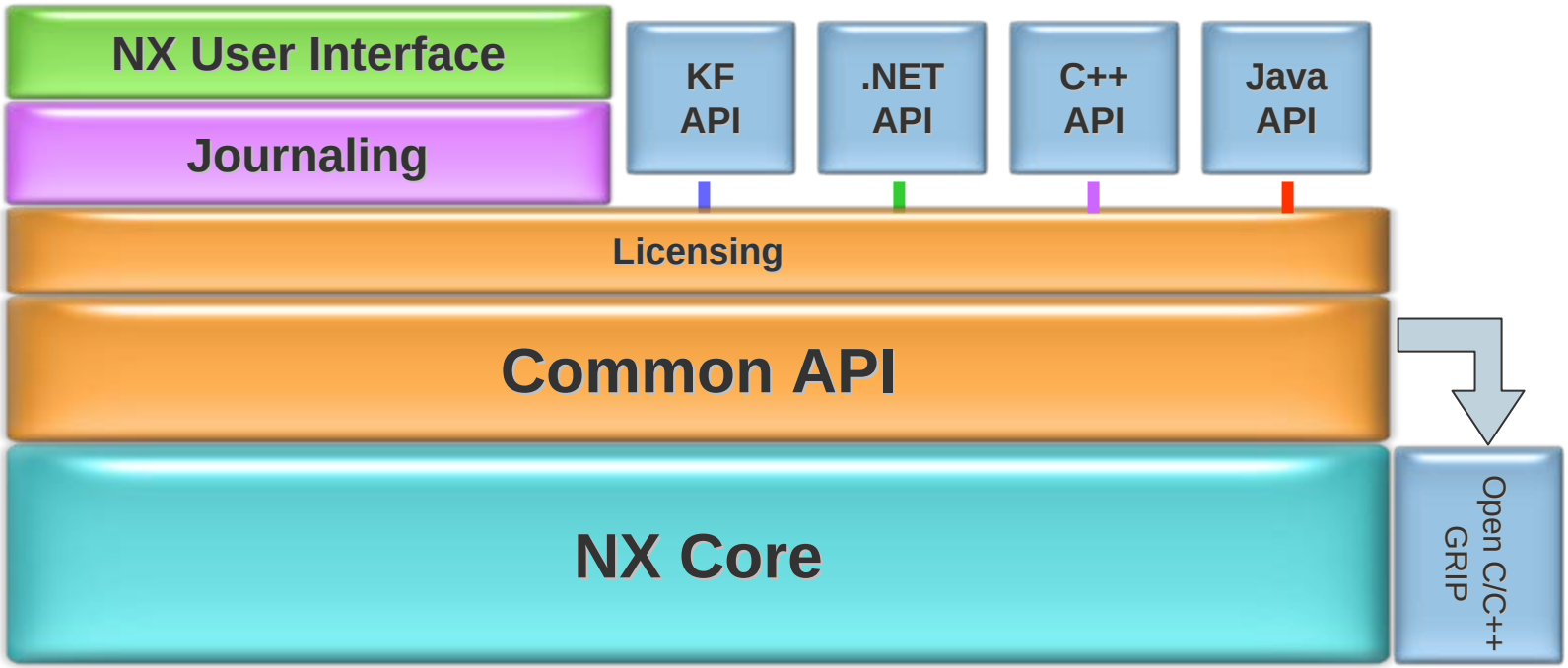
Vision - the best of the best
New NX Architecture (first presented 2004)

- You use the same API used by UGS NX development!!!
- You can program interactively via journaling
- You choose the language, IDE and GUI tools



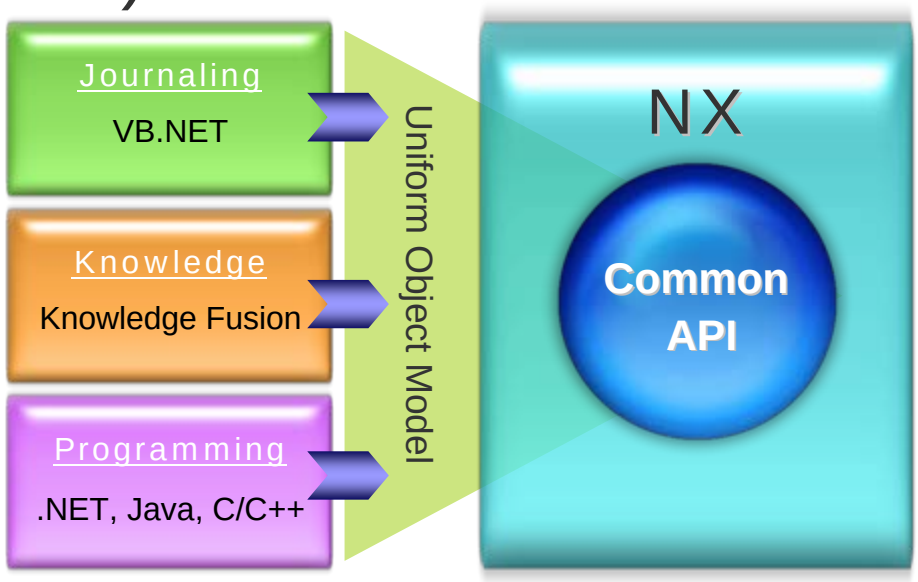
Vision - the best of the best
New NX Architecture (first presented 2004)

- You use the same API used by UGS NX development!!!
- You can program interactively via journaling
- You choose the language, IDE and GUI tools
- **Your investments are preserved**



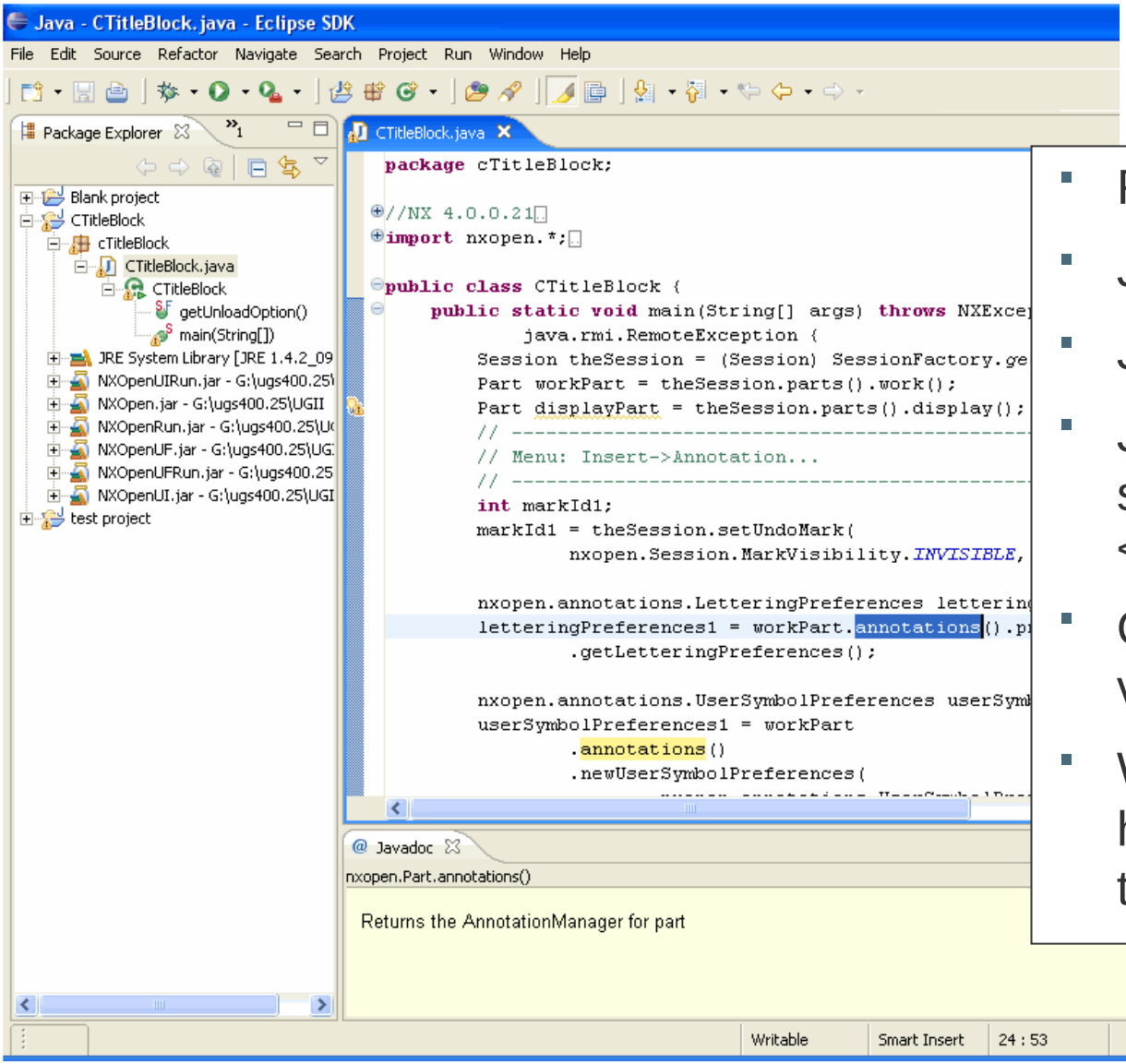
Vision
New Architecture Terms

- Journaling
  your ability to record NX actions using the common API (as of NX 4 you can record in VB.NET, C++ and Java)
- Play or Playback
  your ability to execute a Journal in NX to reproduce your recorded actions (as of NX 4 playback is limited to VB.NET on Windows)
- Common API (a.k.a. JA API)
  consistent set of classes, methods and properties available to all languages supported by NX Open



Common API
New in NX 4

… more languages …



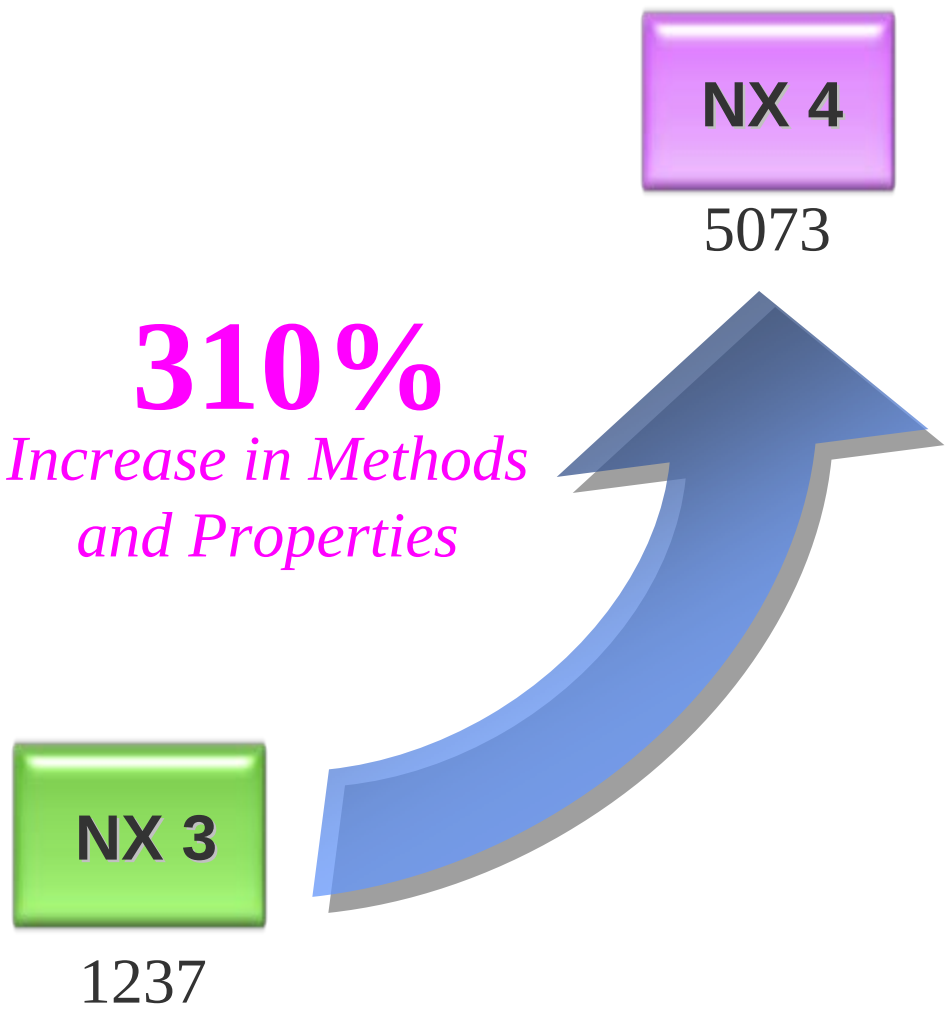
Java

- Full common API support
- Java journaling option
- Java 1.4.2 or higher
- Java runtime environment shipped with NX:
  <NX install dir>\NXJRE\bin
- Obtain an SDK from your vendor of choice
- Warning: if your SDK is higher than 1.4.2 don't use the JRE shipped with NX!

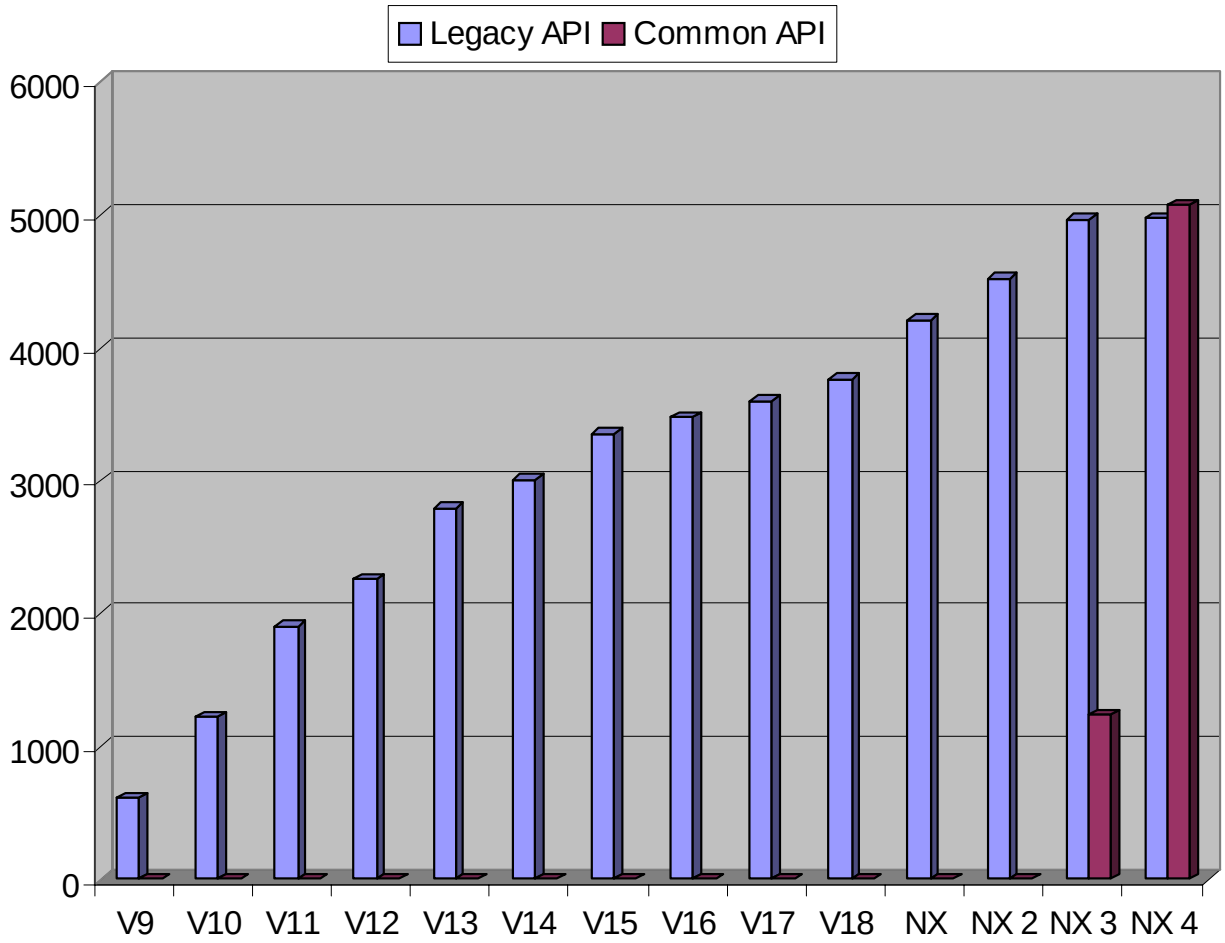
Common API
New in NX 4

… greater coverage …

- New Class Highlights
  - CAE
  - CAM
  - DIE
  - Sheet Metal
  - Geometric Utilities
  - Motion
  - Options
  - PDM
  - Positioning
  - Routing



Coverage
Investment Shifts to Common API



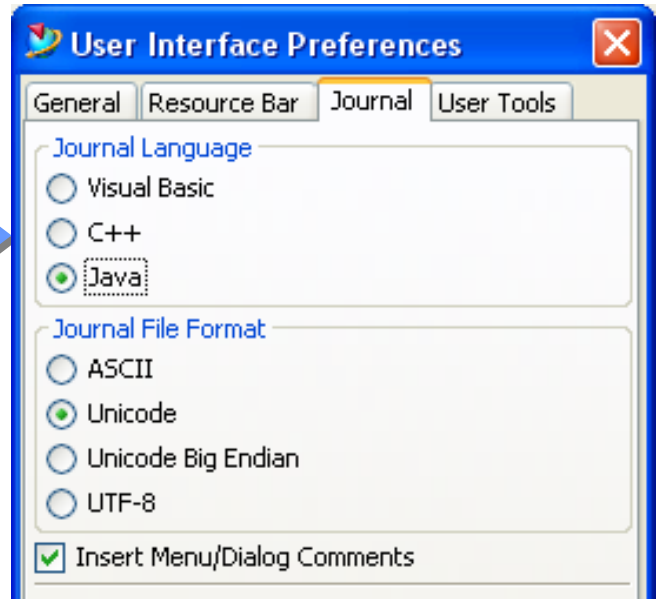
- Legacy API function count by release
- Common API method and property count by release

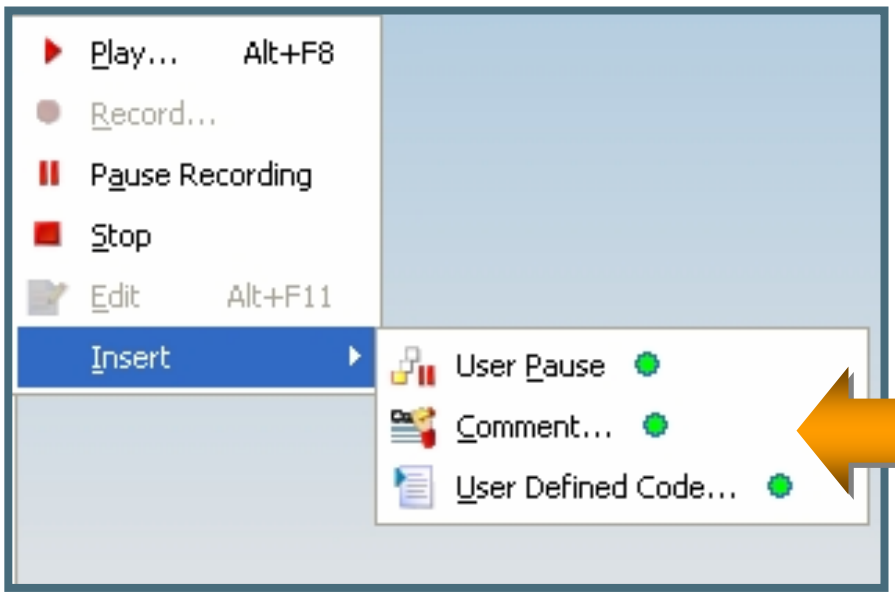
Common API counts do not include
UF wrapper methods or new KF NX classes

Journaling
New in NX 4

… UI improvements …

Select Journaling Language



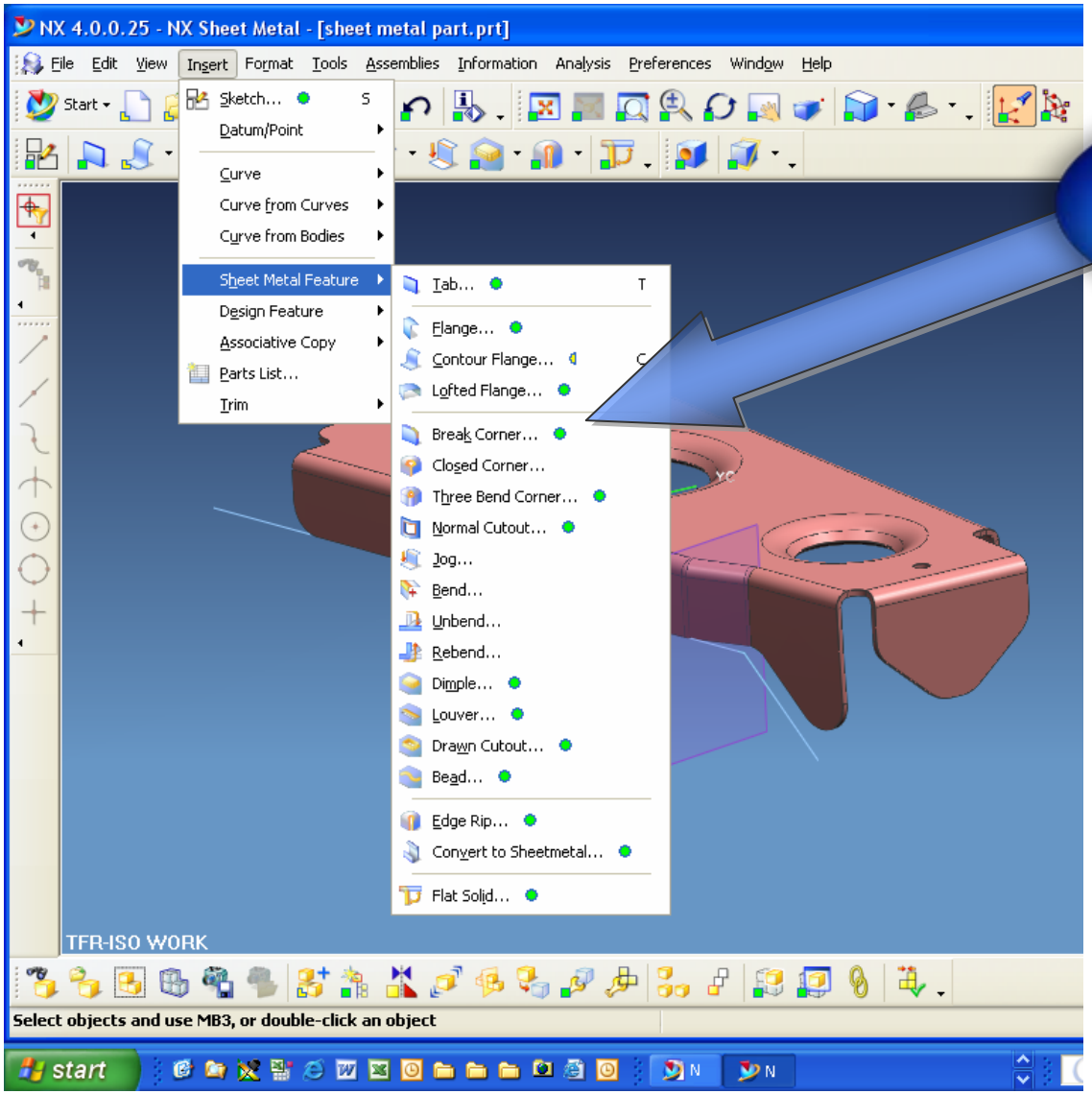


User Insertions

- Pause
- Comments
- Code

Journaling
New in NX 4

… *UI improvements* …



Coverage Indicators

Full Coverage

Partial Coverage

toolbars

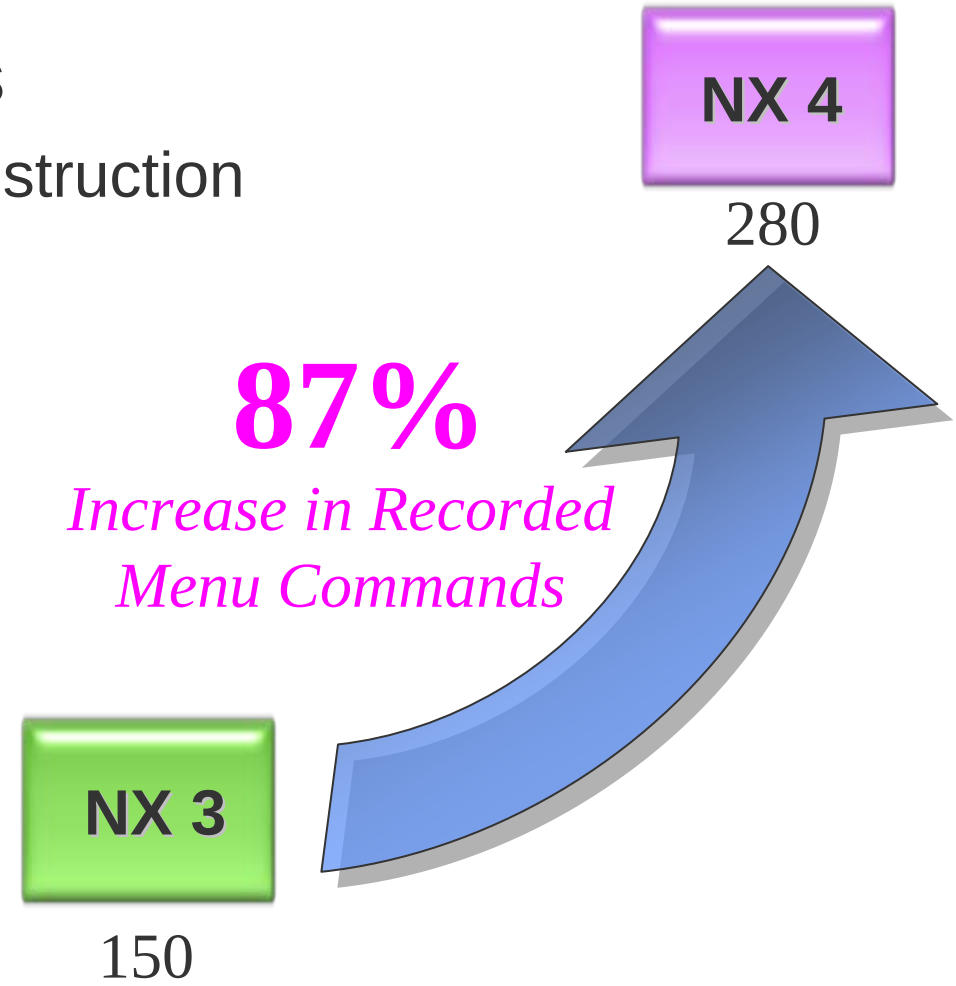
menus

UGII_JOURNAL_INDICATOR=1

Journaling
New in NX 4

… greater coverage …

- NX 4 High Impact Areas
  - Geometry & Feature Construction
    (NX 3: 16 NX 4: 46 288%)
  - CAM (new)
  - Sheet Metal (new)
  - Knowledge Fusion (new)
- NX 3 Focus
  - Gateway
  - Sketcher



For NX 4 many areas of NX focused on providing API coverage, with Journaling to follow in NX 5

Journaling
High Impact Example

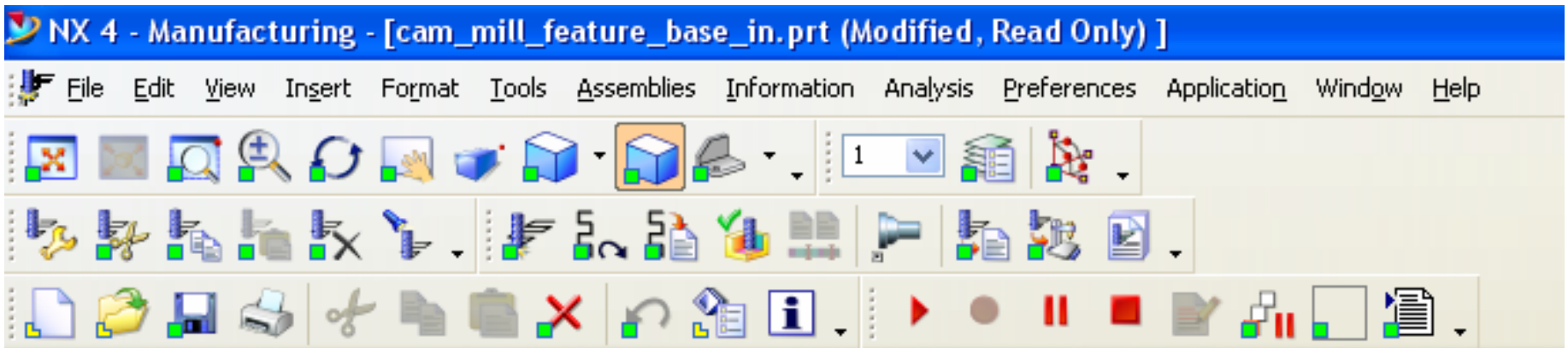
… greater coverage …

CAM Object Creation

- Create CAM session
- Create CAM setup
- Create geometry group
- Create method group
- Create program group
- Create tool group
- Create operation

Actions

- Delete setup
- Delete objects
- Rename object
- Information
- Output CLSF
- Post processing
- Generate tool paths
- Replay tool paths
- Delete tool paths
- List tool paths
- Move objects (Cut + Paste)
- Copy objects (Copy + Paste)
- Show 2D work piece
- Show 3D work piece



Run Time License Control
New in NX 4

- Runtime / Execute Licenses no longer required
- Feature based licenses are now checked
  - common API license requirements are documented
- API methods to permit you to reserve and release licenses (see the license manager class – example to follow)
- Legacy Runtime licenses are still used when available
- Applications must be signed, see:

  NX Open for <language> Programmers Guide :
  Compiling and Running an NX Open for <language> Program :
  Loading .NET Libraries and Licensing
  Running Java Applications and Licensing
  C/C++ not available yet…

GRIP does NOT change. GRIP still requires a runtime license.

Runtime License Control Simple Guideline

Licenses required by an application are those that would be required if a user tries to do the same operations using NX without the application

- General guideline - there may be exceptions
- NX 4 included license checks for enquiry C functions (UF ASK and IS functions), fixed in NX 4.0.1 to only require a Gateway license

License FAQ 1

- What if we already own runtime licenses?

License FAQ 1

- What if we already own runtime licenses?

As long as you continue to pay maintenance, existing runtime licenses will continue to work as before. Existing applications will not have to change, they will work as before using your existing runtime license.

Remember, the common API never used runtime license! If you want to use the new languages, your applications will depend on feature based license checking.

License FAQ 2

- What if we need more runtime licenses because we are not planning to migrate to NX 4 for some time?

License FAQ 2

- What if we need more runtime licenses because we are not planning to migrate to NX 4 for some time?

If you are still running NX 3 (or an earlier version), you can continue to buy runtime licenses.

License FAQ 3

- How do I know what licenses my application requires?

License FAQ 3

- How do I know what licenses my application requires?

1. The reference guides for the common API provide the license required for each method and property.

Remarks
License requirements: solid_modeling ("SOLIDS MODELING")

Text Used by
License Manager

License FAQ 3

- How do I know what licenses my application requires?

2. A spreadsheet is provided for legacy Open C (UF) programs.

license_table.csv

	A	B
1	Required Licenses Rev: 40.24	
2	Funtion	Required Licenses
217	UF_BOUND_ask_number_of_boundaries	(cam_base)
218	UF_BOUND_create_boundary	(cam_base)
219	UF_BREP_ask_edge_class	(solid_modeling)
220	UF_BREP_ask_geometry	(solid_modeling)
221	UF_BREP_ask_identifier	(gateway)
222	UF_BREP_ask_topology	(solid_modeling)
223	UF_BREP_ask_topology_source	(solid_modeling)
224	UF_BREP_attach_geometry	(solid_modeling)
225	UF_BREP_delete_geometry	(solid_modeling)
226	UF_BREP_free_geometry_data	(solid_modeling)
227	UF_BREP_heal_body	(solid_modeling)
228	UF_BREP_make_body	(solid_modeling)
229	UF_BREP_release_topology	(solid_modeling)
230	UF_BREP_validate_topology	(solid_modeling)
231	UF_DRF_ask_ann_arc_seg_angles	(drafting)
232	UF_DRF_ask_ann_data	(drafting)
233	UF_DRF_ask_ann_line_seg_ends	(drafting)
234	UF_DRF_ask_annotation_template	(drafting)

This spreadsheet is found in the overview section of the Open C reference guide. To find a link to the spreadsheet, scroll down to: Initialization and Termination.

License FAQ 3

- How do I know what licenses my application requires?

3. A Perl script is also provided for legacy Open C (UF) programs.

This Perl script is found in the overview section of the Open C reference guide. To find a link to the script, scroll down to: Initialization and Termination.

query_licenses.pl -v myApplication.c
Running in Verbose Mode...
Checking the following files...
seagate_nx_dwg2ps.c
The following function calls were found:
UF_CFI_ask_file_exist() (gateway)
UF_DRAW_ask_drawings() (drafting)
UF_OBJ_ask_name() (gateway)
UF_PART_close() (gateway)
UF_PART_open() (gateway)
UF_PLOT_ask_default_job_name() (gateway)
UF_PLOT_ask_printer_names() (gateway)
UF_PLOT_save_cgm() (gateway)
UF_UGMGR_encode_part_filename() (gateway)
UF_UGMGR_initialize() (gateway)
UF_UGMGR_terminate() (gateway)
UF_free() (gateway)
UF_free_string_array() (gateway)
UF_get_fail_message() (gateway)
uc4561() (gateway)
RESULTS
The following Licenses are Required:
(drafting)
(gateway)

License FAQ 4

- How do I ensure that my application will not encounter license errors in the middle of doing something critical?

License FAQ 4

- How do I ensure that my application will not encounter license errors in the middle of doing something critical?

Using the LicenseManager class, explicitly reserve the required license when your application starts.

```
theSession.LicenseManager.Reserve(“solid_modeling")

… your application …

theSession.LicenseManager.Release(“solid_modeling")
```

License FAQ 5

- How do I upgrade my existing author licenses to the common API?

License FAQ 5

- How do I upgrade my existing author licenses to the common API?

1. Your existing licenses will still work. You do not need to upgrade.

2. As of NX 4 two author options exist:

a) NX Open Toolkits Author (C/C++, .NET and Java)
b) NX Open for .NET Author (.NET only)

3. Upgrade products exist for all existing author combinations. Upgrades are not free but are heavily discounted. See your sales rep for details.

Other NX 4 Topics

- Open C changes
  - 108 PRs Closed in NX 4
  - 168 new functions
  - 168 obsolete functions (116 for Routing RLIST)
- New Documentation
  - NX Open General Programmer’s Guide
  - NX Open Java Programmer’s Guide
  - NX Open Java Reference Guide

Other NX 3 and NX 4 Topics
High Priority Additions (NX 4.0.1 and NX 3.0.4)

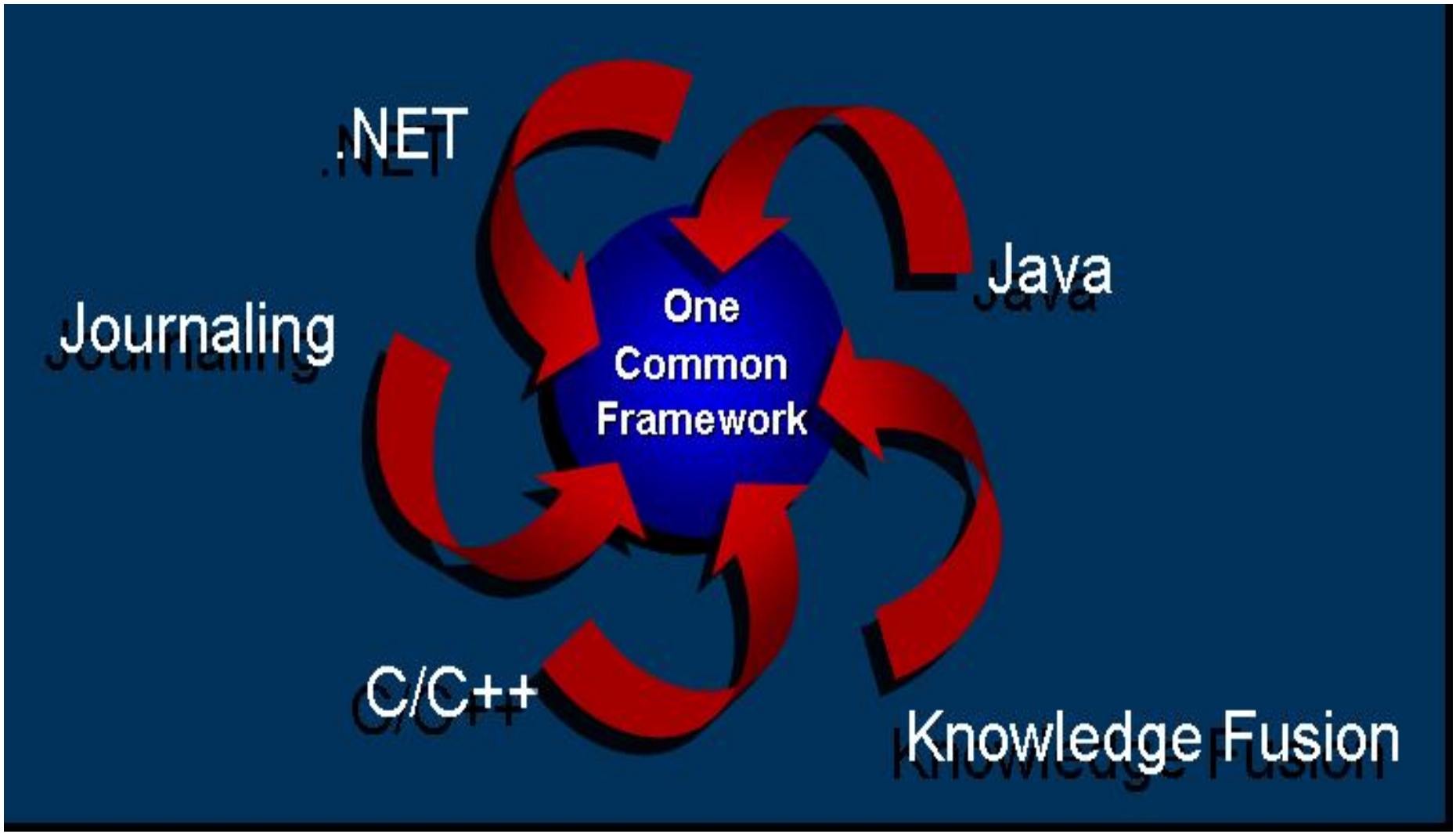
- UFTrns class to replace uf59xx transformation functions
- Additional UFUi methods to replace legacy dialog functions
- UFPatt methods to replace pattern functions

AskLastPickedView	uc1653
AskStringInput	uc1600
DisplayMenu	uc1603
DisplayMessage	uc1601
DisplayMultiSelectMenu	uc1605
PickCsys	uc1630
PickPoint	uc1615
PickView	uc1652
PointSubfunction	uc1616

CreateCsysMappingMatrix	uf5940
CreateReflectionMatrix	uf5946
CreateRotationMatrix	uf5945
CreateScalingMatrix	uf5944
CreateTranslationMatrix	uf5943
MapPosition	uf5941
MultiplyMatrices	uf5942
TransformObjects	uf5947

AskData	uc5822
Import	uc5823
CycleErrors	uc5824

End of NX 4 Update



How to Move Forward

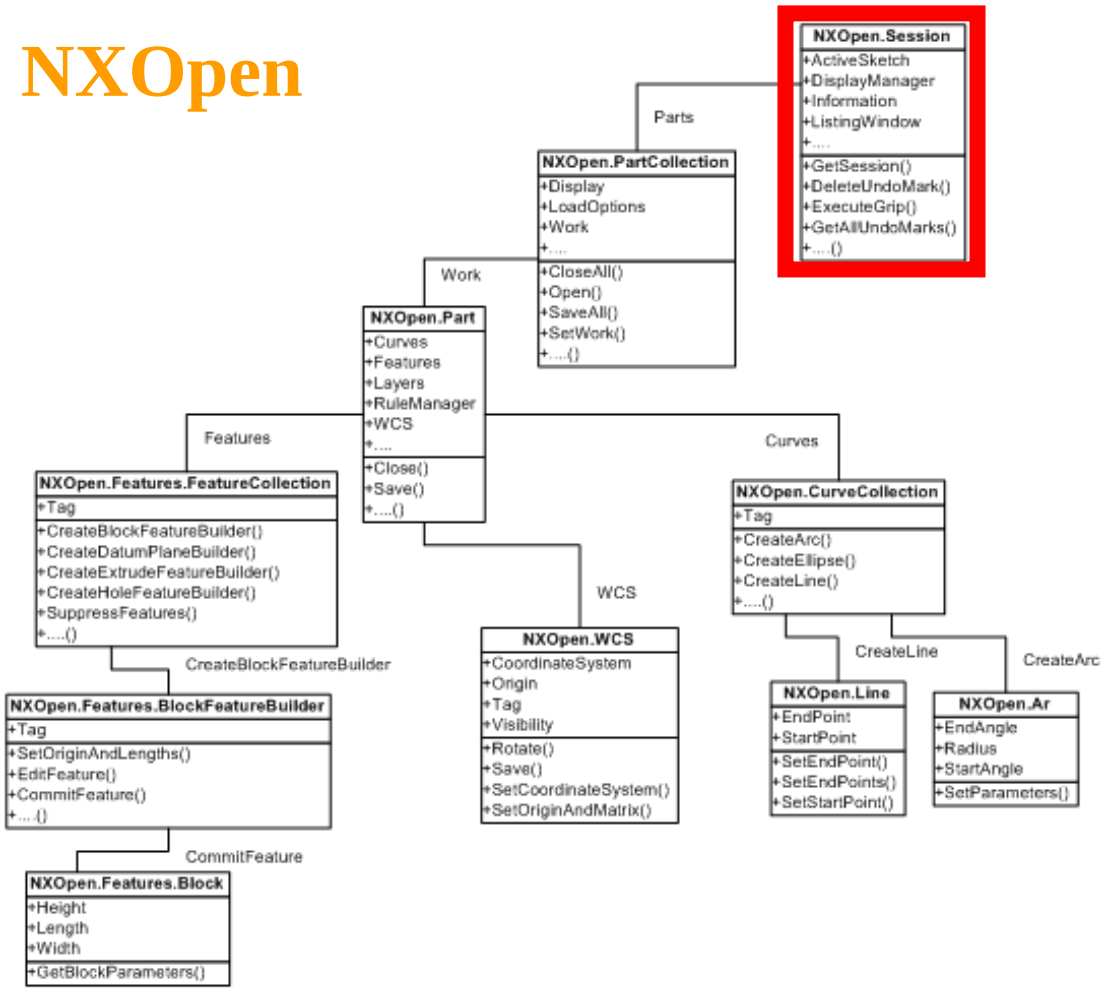
- Learning the common API
  - getting started
  - class hierarchy fundamentals
  - exploring the examples
- Turning Journals into Applications
  - selection and data entry example
  - debugging a journal
- Using Your Old Open C/C++ Programs
  - using existing modules with new applications
  - full integration between old and new code
- Migrating from Open I-deas
  - application mapping tool
  - program file mapping tool

Learning the Common API
Getting Started

- Journal something
- Analyze the source
- Look up what you don’t understand
- Review the examples
- Start small and grow…

Learning the Common API Class Hierarchy Fundamentals

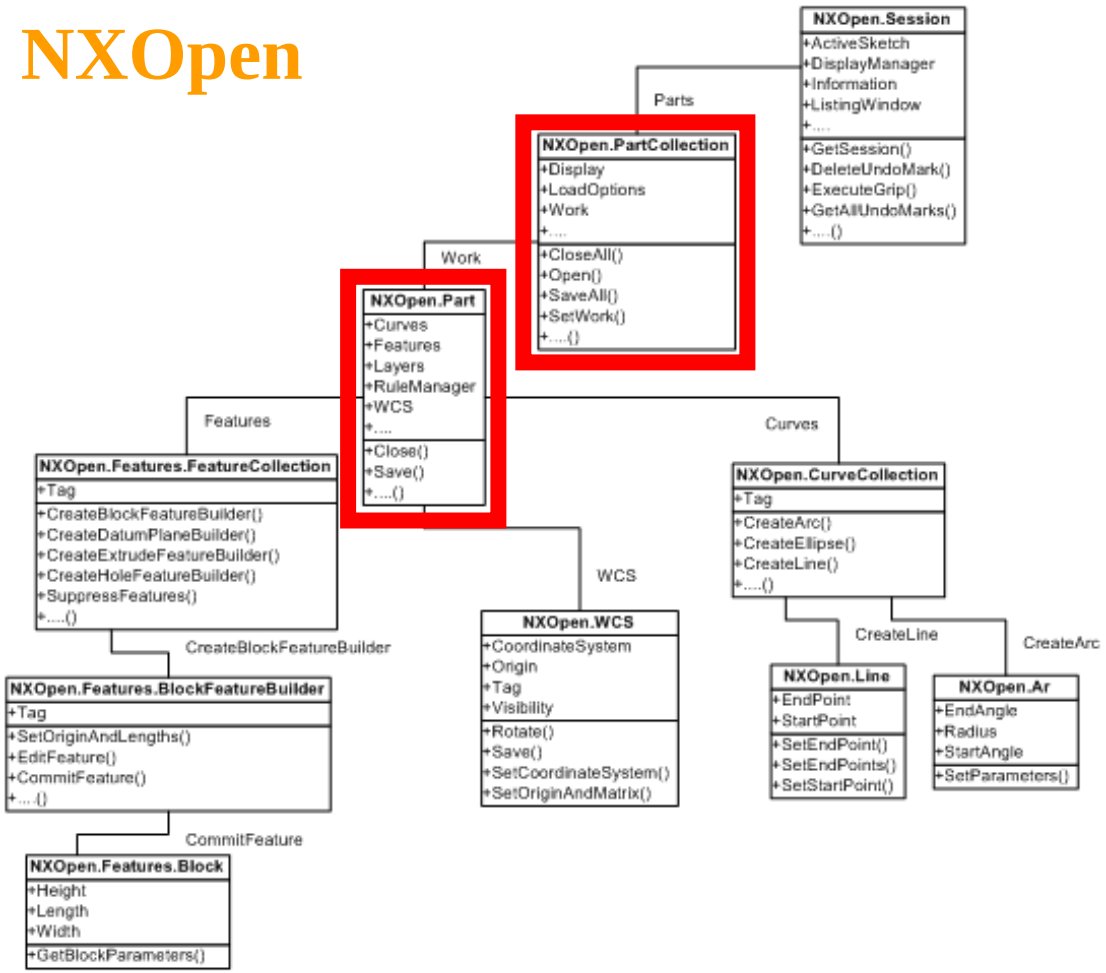
NXOpen



- Session
  - Serves as the "gateway" class for most other classes
  - References to most other classes in the common API are accessed directly or indirectly via methods and properties of this classes
  - Session object is analogous to the Document class in the Word SDK
  - You should obtain a Session object first

Learning the Common API Class Hierarchy Fundamentals

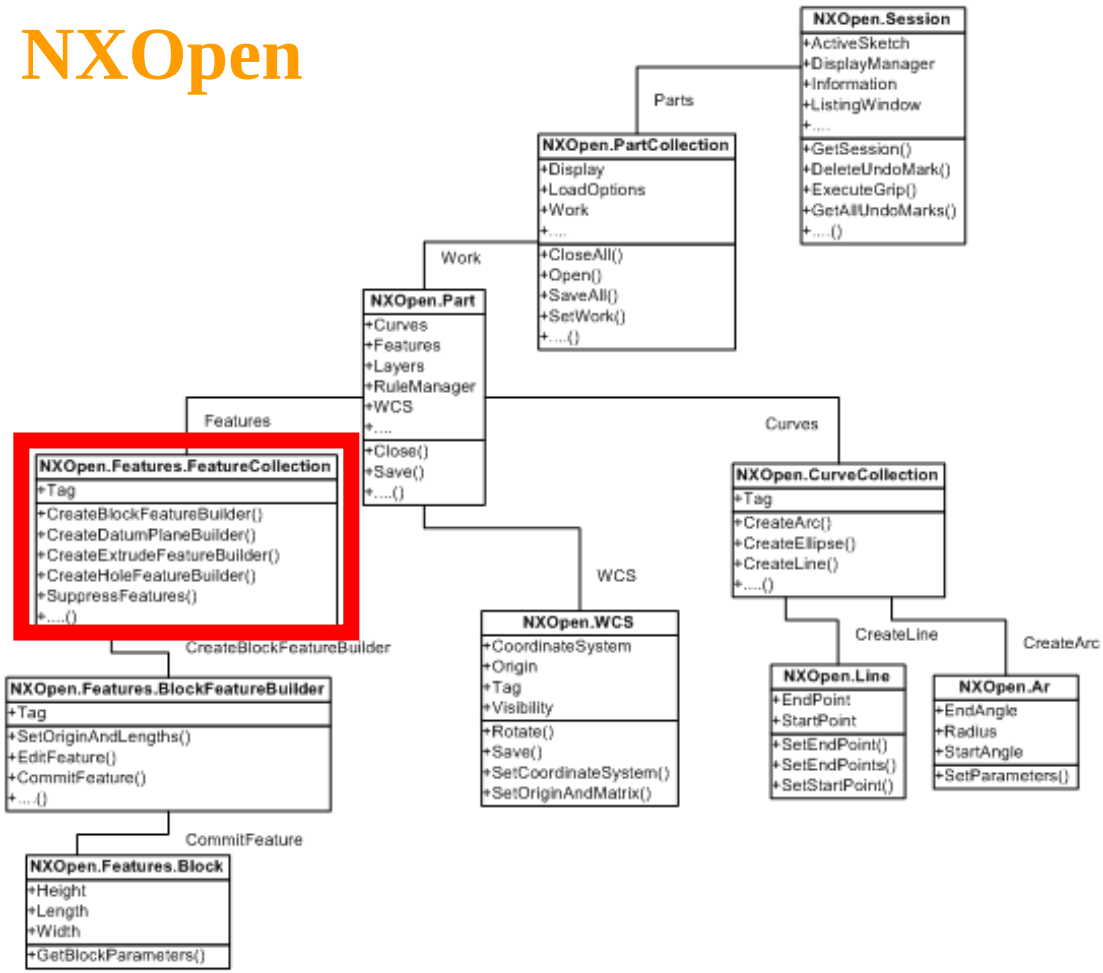
NXOpen



- PartCollection
  - access the parts in a session, open parts, create new parts, …
- Part
  - Methods and properties for a specific part
  - Provides access to…
    - geometry, bodies and features
    - drawings
    - CAM setup
    - display settings
    - layouts, views
    - …

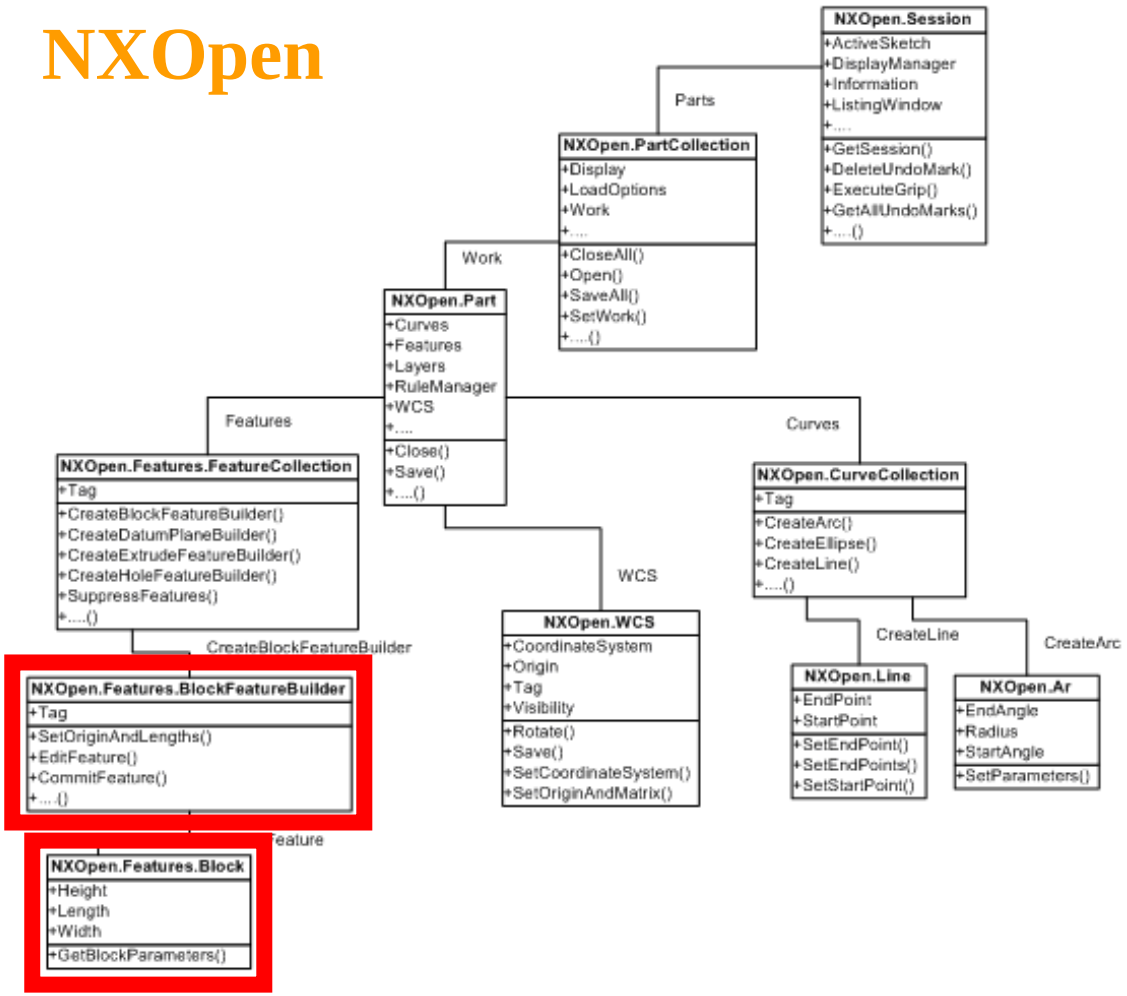
Learning the Common API Class Hierarchy Fundamentals

NXOpen



- Features
  - part property
  - also a name space
- FeatureCollection
  - manages the creation and editing of tagged objects like curves, bodies and drafting objects
  - CreateXyzFeatureBuilder methods create specific feature builder objects for new or existing features

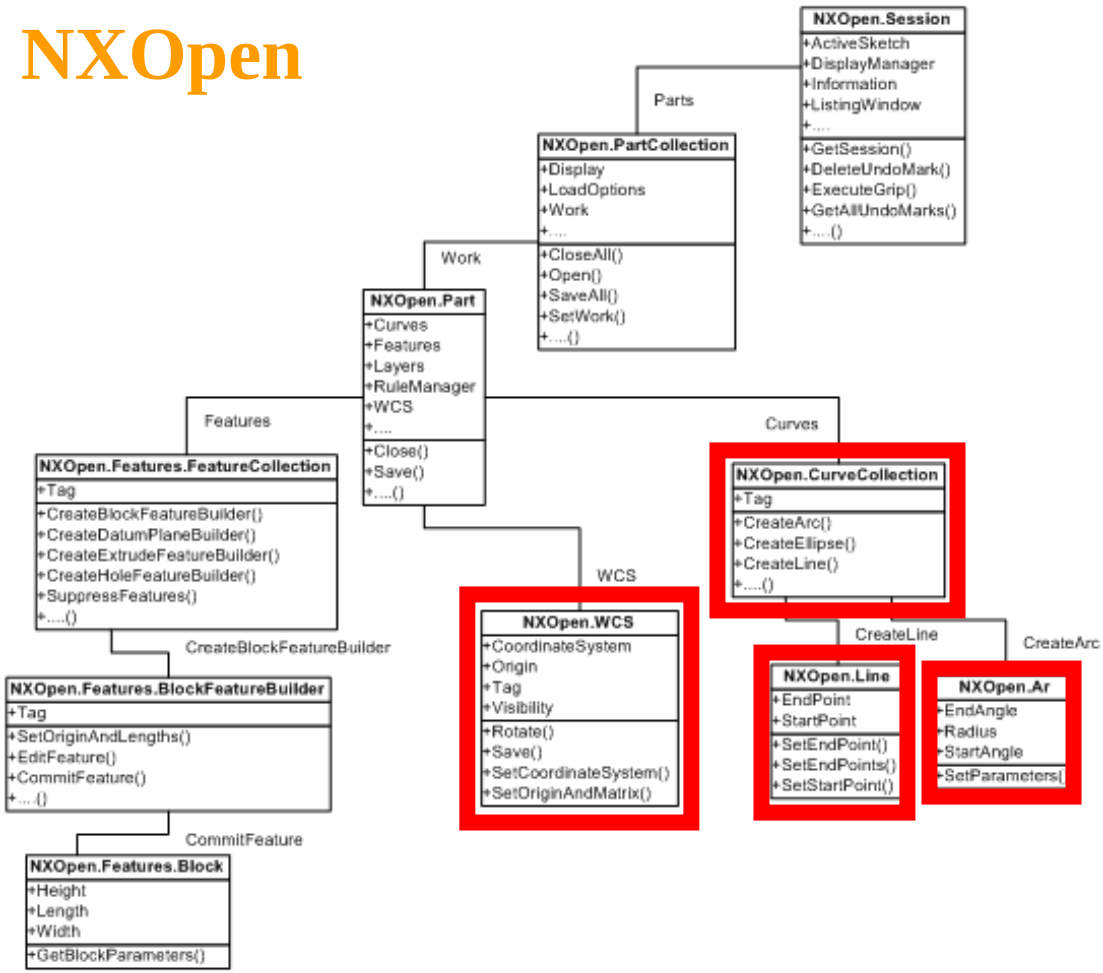
Learning the Common API Class Hierarchy Fundamentals



- <xyz>FeatureBuilder
  - manages the creation and editing of a specific type of feature in a part
  - the Commits methods creates or edits a specific feature in the part
- Active Feature Specific Classes
  - query feature information for an active feature in the part
  - access bodies, faces, edges (cast down to a BodyFeature – not shown)
  - access feature properties

Learning the Common API Class Hierarchy Fundamentals

NXOpen



- Other Part Properties
  - CurveCollection
    - class for specific curve types (line, arc, …)
  - WCS
  - …

Learning the Common API Class Hierarchy Fundamentals

Other Notable Classes

- UI
  - the "gateway" class for all NX user interface classes, including the SelectionManager
- UFSession
  - access to wrapped UF functions

Learning the Common API
Exploring the Examples

- Many sample programs are shipped with NX
- Examples for all languages are included
- Great for getting started or for future reference
- Remember, use Journaling to interactively create your own samples!
- Explore the samples

<NX install dir>\UGOPEN\SampleNXOpenApplications

Turning a Journal into an Application Selection and Data Entry Example

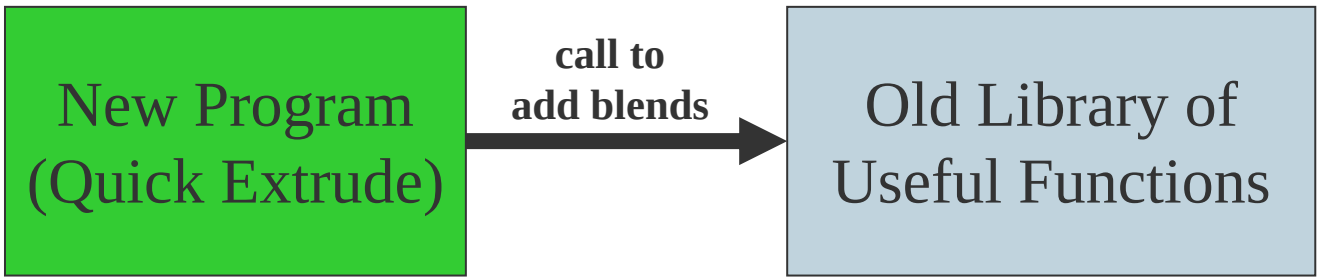
- Journals are created using the specific objects selected by the user
- This is required to automate the specific steps
- Selection and other user entry code may be added to let the user select other objects
- Demo steps using the Quick Extrude example

Turning a Journal into an Application
Debugging a Journal

- First convert the Journal into an automation program
- Create a Visual Studio project
  - VB project/class library
  - build the VB file into a managed DLL
  - Add GetUnloadOption method
- Start VS and attach NX .exe to debugger
  - Project Properties → Configuration Properties → Debug
  - Start External Program set to: ugraf.exe
  - Set breaks in VB journal
  - Debug → Start (will start NX)
  - Run DLL via File → Execute → NXOpen…
- Demo using previous example

Using Your Old C/C++ Programs
Using Existing Modules with New Programs

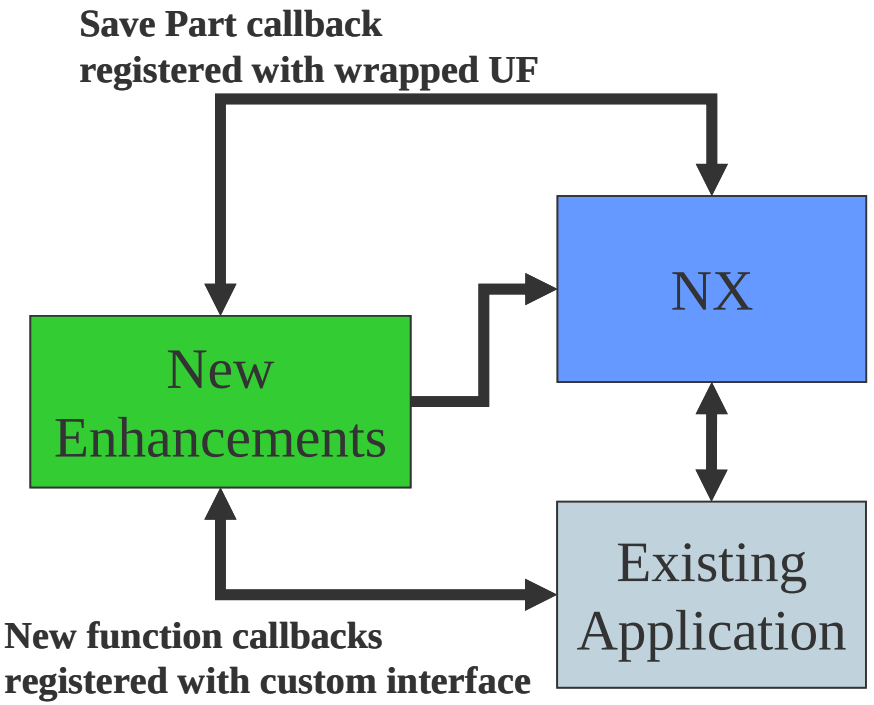
- This demo will call an existing C function to add blends to the edges of a face selected by the previous Quick Extrude example



- Demo accessing and passing arguments to an existing (C/UF based) DLL from VB.NET

Using Your Old C/C++ Programs
Full Integration Between Old and New Code

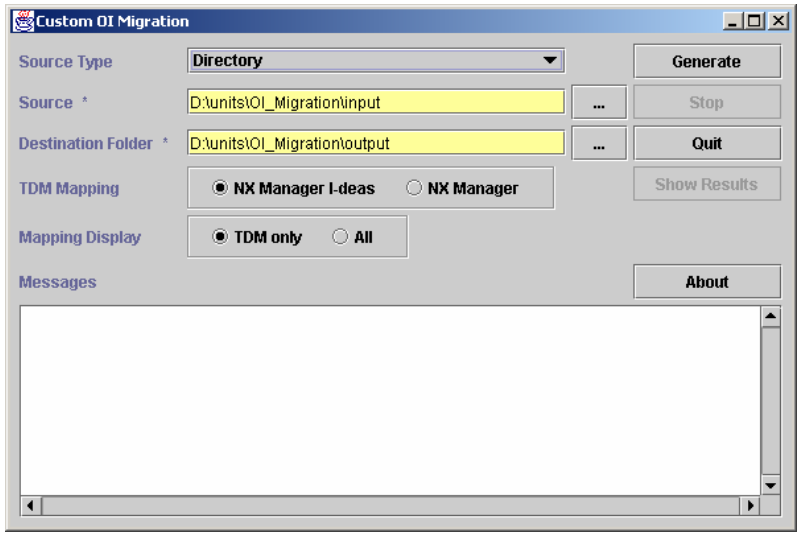
- This demo adds a button to an existing UI Styler dialog, the button is handled by a method in a VB.NET module
- It also shows a wrapped UF function used to register a Save Part callback that is handled by the same VB.NET

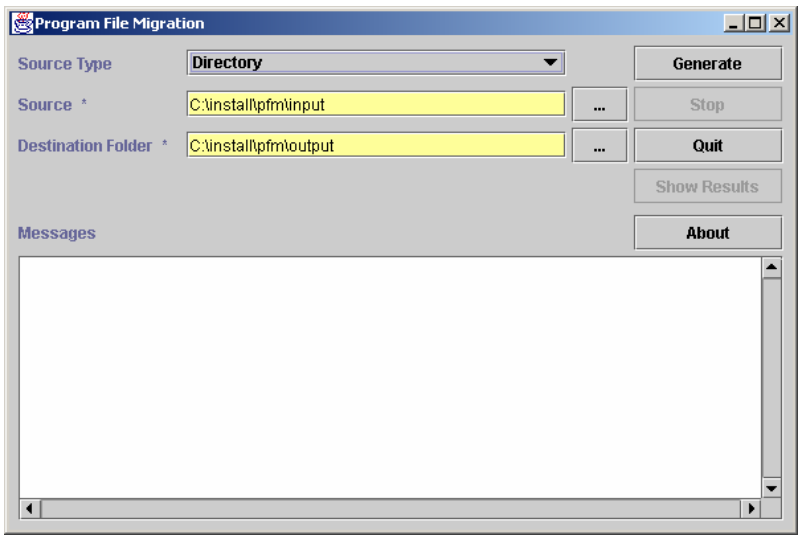


- Demo UF wrappers, custom interfaces and callbacks

Migrating From Open I-deas Mapping Tools

- Open I-deas Mapping Tool
  Java app provides mapping from Open I-deas methods to NX Open, Open C (UF) and ITK
- Program File Mapping Tool
  Provides a mapping from I-deas Program Files Commands to NX Menu Commands
- Both tools ship with I-deas starting with ID11 1Q04
- For more information please attend:
  - I-DEAS TO NX: TOOLS UPDATE
  - Wednesday, 1:45
  - Rick Schnelle (UGS)

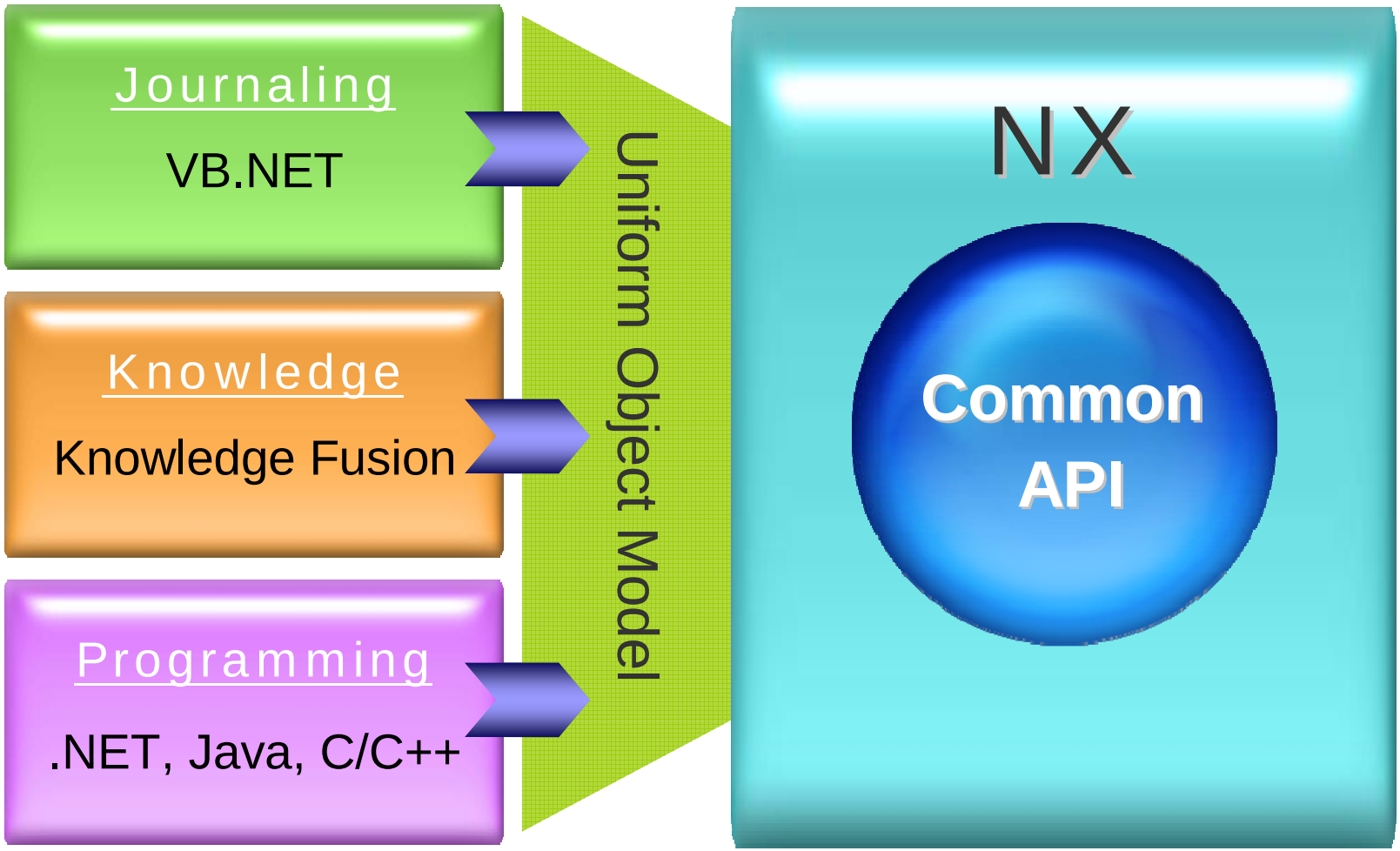




Questions?

Q & A

End
Thank You



Abstract

The next generation NX automation tools were introduced at last year's PLM World. UGS has made great progress towards delivering this vision. For those who missed it, we will quickly restate the journaling and common API strategy. Then we will review specific progress made during 2005. For instance, the new Java journaling and programming option will be demonstrated and the new NX 4 license strategy will be discussed. Several of the demonstration applications shipped with NX 4 will be reviewed and topics such as interoperation between the legacy APIs and the new common API will be discussed. If your job concerns automating NX processes then attending this session is essential.