

Introduction to NX System Administration on Windows

Ken Akerboom

Moog, Inc.

East Aurora, NY

kakerboom@moog.com

(716) 687-7939

Yes, I'm still a part of:

Specialty Engineered Automation

akerboom@sea4ug.com

www.sea4ug.com

Does your brain feel like this?

Insert
Cartoon

Notes on This Seminar

- Many slides have additional info in “notes page” text (“*” added to slide title)
- While I know (sort of) what TDM is, this seminar has nothing to do with migrating TDM (or any other PDM issue)
- While this is “MS-Windows centric”, Unix users should still be able to learn something
 - I will try to point out Unix vs. Windows issues
- NX2 slides have been removed
- Some changes from CD – will be updated on www.plmworld.org web site.

Who am I?

- Unigraphics user since 1984 or so.
- I started with UGI (no, not UGII version 1) on a VAX 11/780 with *ONE* MB RAM, running two D-100C terminals (yes, we spent the extra \$10,000 *each* for COLOR), a HP pen plotter, and a VT100 MUXed over a 9600 baud modem (see www.plmworld.org ▶ museum link)
- I have been a user, system admin, and automation programmer ever since.

The Best Thing About NX:

- There are often several ways to do any task

The Curse of NX:

- There are often several ways to do any task

Goals for this Seminar

- You will have a basic understanding of how to install, customize, and manage, your NX system.
- You may not know HOW to do something, but you will know something CAN be done, and how to find more info about how to do it.
- These are MY recommendations. I strongly suggest you look at others' methods (e.g. Rob Cohon's posts on the BBS) and pick and choose the techniques that best fit your company's needs.
- YMMV!
- I expect to learn something today
- Ask questions as we go along!

Seminar Organization

- Basics of NX (versions, etc.)
- Licensing
- Install, Uninstall, & Upgrading
- Customizing your configuration
- Printing/Plotting
- Where can I get help?
- Miscellany
- Automation Tools (Programming)

NX Release Scheme/Naming

- Major releases [now] called NXn
 - NX[1], NX2, NX3, ...
- Maintenance Releases **USED** to be called QRMs (and IRMs)
 - MRs are sort of equivalent to a MS Service Pack.
 - All customers will get CD [I think...]
- There are also Maintenance Packs
 - A limited number of critical fixes
 - Specific to a certain MR
 - Probably NOT sent out on CD to customers (unless they ask for a CD)

NX Software Version Numbers

- NX3 example
- “Pre-release” (e.g. beta) code referred to by “phase” (e.g. NX3 phase 20)
- 3.0.0.21 = Official Release
 - 3.0.1.3 = first MR
 - 3.0.1.1, 3.0.1.2 NOT released to general user base
 - 3.0.2.x = second MR
- As MPs are specific to a version, they are numbered consecutively **FOR THAT VERSION**
 - 2.0.4.2 MP7
 - 2.0.5.2 MP3, 2.0.5.2 MP4
 - 3.0.0.21 MP1

Issues with FlexLM and license files

- Before you install the license server, you need a Flex license sheet (which needs hardware info)
- NX2 License server (only) can still be installed on a Windows NT system (I don't know about NX3-4)
- The NX License Server must be running to run any NX application (but installs do NOT require it)
- The license server can be a later version than the rest of the NX software
 - An NX4 server can serve licenses to a NX2 client
- The other way does NOT work
 - An NX3 license server cannot serve licenses to an NX4 client

FLEX Server Configurations

- Can have just one server (typical case)
- Can have “redundant” configuration
 - 3 servers, one SHARED license file
 - All licenses are available as long as 2 or more servers are available.
- Can have multiple “independent” servers
 - Each server has unique (non-shared) license file
 - Client ENV points to 2 or more servers
 - Can be more than 3!
 - Allows license “overflow” (if one server is out of a license, client tries the other servers)
 - If a server goes down, all its licenses go with it
 - NOTE: If first server does not have license, or is not up, you can get long delays

Flex License Sheet

- Ethernet MAC address (or dongle) required to get one (unix: see docs on how to get this)
 - Windows: `C:\> ipconfig /all`
- Important stuff in header [below is faked]:

All contacts with GTAC require your "Sold to" ID

Web based support tools require WebKey, Use this code to get one

```
#####  
#                               Unigraphics Solutions, Inc.                               #  
#                               License File                                           #  
#                               Sold To: 1012345                                       WebKey Access Code: ABCDEFGHIJ #  
#                               Contact Name: KEN AKERBOOM                             Created: 10/26/2002           #  
#                               Customer Name: SPECIALTY ENGINEERED AUTOMATION          #  
#                               E-mail Address: akerboom@sea4ug.com                     Version: 1                   #  
#####  
SERVER YourHostname 08002ba6010d 27000  
VENDOR ug1md  
PACKAGE ADV_MFG ug1md 19.0 34DOABB4B00A \  
COMPONENTS="ADV_MFG_assemblies ADV_MFG_cam_base \  
"
```

Use Email Distribution List?

This license for NX1

IP Port for INITIAL Flex Network Communication

Server Host name (EDIT!)

Server MAC address

Flex info also in Syslog

- Help ► NX Log File
 - For some reason these are backwards on NX 3.0.0.21 & 3.0.1.2:

```
UGFLEXlm Log: License server "27000@YourHostName" has been connected.  
***** NX Licensing Information *****  
Server ID : ABC123  
Webkey Access Code : 1012345 - SPECIALTY ENGINEERED  
*****
```

IP Port@server for FlexLM

WebKey access code

"Sold to" ID

Notes about Flex Setup*

- As long as client can “ping” the license server by name, licensing *should* work:
 - `C:\> ping yourserver`
- You MAY (if “ping” doesn’t work) need to add the license server to your client’s hosts file:

```
C:\WINNT\system32\drivers\etc\hosts
```
- Flex server runs as a Service in MS Windows (as a daemon in Unix)
- If you’ve NEVER installed NX before, check TCP port 27000 isn’t in use by another application (there seem to be several spyware apps that use 27000!)
- If you want a better front end on the flex utilities, check out:
 - <http://freshmeat.net/projects/phplicensewatcher/>
 - See full post from BBSnotes in “notes” below

FlexLm, NX and Firewalls*

- The listed port (default 27000) only used for initial message, other ports used for subsequent packets.
- Firewall - If you “allow” by specific applications (vs. IP addresses), there are LOTS of NX apps – ugraf.exe, each translator’s .exe, ugpc.exe, etc. that need to be “allowed” (and FlexLM service & possibly plotting).
- All need to be allowed to both send & receive.
- McAfee Virusscan v7 incompatible with FlexLM (see PR-04820797)
 - Virusscan v8 fixes this
- Note MS Windows XP SP2 has the built-in firewall enabled by default
- Also see:
 - http://ftp.ugs.com/cert_reports/1066.pdf
 - http://uganswer.ugs.com/gqcol/flexlm_firewall.PDF

Before Installing NX

- Read the release notes (less “RTFM” responses☺)
 - OS Versions (and patches/SPs) required
 - Hardware obsoleted
 - (if NX Open development) Version of C/C++ compiler
 - (or VB or VC# or J# or Java or...)
 - Other notes/caveats/etc.
 - OS plans for next release (preliminary)
- Also read the “What’s new” guide
 - Sometimes impacts environment variables

Installation of NX (client)

- You **MUST** start with a major release
- You can then install the latest Maintenance Release
 - Maintenance Releases are cumulative (if you want to install NX 3.0.4.x, you don't have to install 3.0.1.x and 3.0.2.x and 3.0.3.x)
- Same with Maintenance Packs
 - They are cumulative for that MR (if you want NX 2.0.5.2 MP6, you do **NOT** have to install NX 2.0.5.2 MP1, MP2, MP3, etc.)
 - To me, unless an MP fixes a bug critical to MY installation, I wouldn't install MPs

Installation of NX (2)

- Install on a local disk vs. shared disk on a server
 - When disks were 2GB, this was an issue.
 - Now, with 40+ GB disks standard, I'd install everything locally (for less network traffic & better performance), except (maybe) the *CAST FILES* (and maybe UG Docs – install time).
- Again, as size is much less of an issue, I'd install **EVERYTHING** in the distribution on each workstation (except, if you don't use it, UGManager).
 - If you buy a new license for something you didn't install it, you'll have to reinstall NX, AND re-install the MR (AND the MP!).
 - For NX2 and earlier, I wouldn't install the plot queue manager everywhere either (just make sure it is installed somewhere!)
 - For NX3 & later, note the SDI plotting software **MUST** be installed on every system!

Installation of NX (3)

- One issue with CAST is you can add custom courses, which may be easier to manage if the entire CAST tree is on one share that everyone uses.
 - Use “Configure only” option on clients.
- Note the CAST CD is sent after the main distribution (a couple months or so)

Software from the Web

- While there is A version of Postbuilder on the CDs, the most recent version is available on the web (“Downloads” link off GTAC home page).
- I believe some other modules also have updates this way (e.g. MoldWizard)
- MRs and MPs available on web FIRST (CDs later)
- MPs only distributed on the web.
- Beta software ONLY distributed from web *
- Can get full releases from the web * (CDs later)
- * requires you to register the download IP address
 - This is required for export regulations

Other Install Notes

- Some MRs/MPs had problems leaving files with the “Read-Only” attribute set. This caused problems with later MRs. Fix:
 - In Command prompt:
 - `C:\> CD /D "C:\UGS\NX 3.0"`
 - `C:\UGS\NX 3.0> ATTRIB -R .* /S /D`
- Generally, I move any “Uninstall” start menu shortcuts to an “Admin only” uninstall folder
 - So users don’t “accidentally” run them
 - I don’t think UGS does this anymore

Other Install Notes (cont)

- It seems to work OK if you install MS Office AFTER installing NX (Excel is used for part families, and [for some releases] tabular notes)
- If you do NX Open development, you probably want to install Visual C++ (VB/C#...) before you install NX (so the NX Appwizard gets installed).
 - If you didn't, see UGSolutions article 001-4892591

My typical client install sequence

- Install NX Base + [almost all] Options
- Install all Translators
- Install the On-line Docs
- Install CAST (files local or on server)
- Install Maintenance Release
- Install Post Builder
- Document the install
- Start customizing
 - DOCUMENT your customizations, or make them “self documenting”

Automating Installation

- Cloning a entire disk/partition
 - Ghost et. al.
- Cloning just the install of NX
 - msi/SMS/Active Directory
- Automating the regular install
 - Windows Installer
- Automating application of customizations

Cloning a Disk (e.g. Ghost)

- **Good if:**
 - You have limited variety of hardware/software
 - So you can build a (or a few) “master” systems
 - You are already doing this for other reasons
 - Side benefit – “back up” of COMPLETE system (Windows & all software)!
- **Not so good if:**
 - Wide variety of configurations (hard/software)
 - Just want to install/upgrade NX

Cloning the Install of NX

- msi/SMS/Latest version of Ghost/Active Directory
- Good if:
 - Understand mechanism
 - Just doing an “upgrade”
 - Have a system to “play with” (to build install)
 - Have the software (SMS/Ghost/etc.)
 - Same install can be used on different hardware configs (but all have to have same OS & partitioning)
- Bad:
 - Can overload network if trying to install on too many clients at the same time
 - Can be hard to set up (it may take a few tries to get “right”)
- Results in a “NO QUESTIONS” install of (potentially) several things at once

Cloning an Install - Method

- Create “virgin” system (Windows+SP, *maybe* network & disk drivers) – as few DLLs as possible
- Capture “baseline” (the “before” picture)
- Install software (NX + Translators + etc.) and reboot
- Capture “Differences” file (what changed: files, registry settings, INI files, etc.)
- Now “Differences” file can be applied to another system
- Typically a configuration file – what to include/ignore in the “differences” file
 - Hardest part of whole thing – may have to do several times to “get it right”
- Test the differences file – does it install the software?
 - Does the installed software actually work?

Cloning Install - Hints

- My typical “differences” file had ONLY “standard” NX – NO customizations.
 - Too many distractions
 - may capture debug of mail problem
 - May not know what I want to customize!
- Customizations done later (will be covered in a bit)
- Building MSI scripts – See Minasi
- SMS – Product from Microsoft
- Latest version of Ghost can (supposedly) do this.
- Check if & how file protections are captured

Automating the Install – NX3

- Uses Windows Installer technology
- Install CD – “View Documentation” “NX Installation Guide”
 - Or CDROM:\Docs\Unigraphics.PDF
 - Section “Automatic Installation of Unigraphics NX3”
 - Needs an MSI file
 - Default = CDROM:\nx030\NX 3.msi

Automation of Customizations

- In general, I do the customizations (e.g. ENV settings, defaults, and other stuff) separate from the regular NX install.
- Generally I write a .Bat procedure to ensure they are done correctly and consistently on all workstations.
- Note if most customization is done to files in a shared area on a server, you don't have much to customize on the client...

Running multiple NX versions at the same time (concurrently)

- Install (including Maintenance Releases/ MPs) from oldest to newest
- Use a .Bat (or Start Menu) to start specific versions
- Unless you edit file types in Explorer, double clicking on a part file will open the LATEST version of NX
- Note uninstalls delete some ENVs, etc., so you have to re-apply them (see next slide)
- Each new install re-defines the OS level ENV variables
- Installing a new MR/MP in a concurrent environment
 - Make sure UGII_BASE_DIR points to correct version's top-level folder (see UGSolutions article 001-4596995)

Un-Installs*

- In multi-version installs, un-installing an earlier version will remove all the system ENV variables.
 - For NX1 and earlier, re-run the setup program and do a “configure only” install
 - For NX2, re-run the install and do a “repair”
 - May require the CD
 - May break UGII_ENV_FILE system ENV, requiring manual re-definition
 - See notes page for reference
- See Notes page for Excel fix

Upgrading

- Note macros may be broken in new release
- TEST CAM
 - Develop “Test Suite” of parts, operations, and “current version’s” output (CLS and/or posted code)
 - Posts – re-post existing test operations/files & compare
 - Did posts break?
 - Re-generate test ops, post & compare
 - Did generation change?
 - Create new ops, post and compare
 - Did the op templates change?
 - **I cannot emphasize this enough: TEST CAM!**
 - Test as early as possible, so if any bugs/problems, you can get them fixed in a Maintenance Release, or at least have time to develop & implement workarounds

Running NX w/VPN access to flex server

- Make sure the “ping Flex_Server” test works with VPN up.
- Serving just licenses works OK over 56kb modem
- Don't try to run NX itself (the software) over the VPN – Install NX on the “remote” system.
- Even part files will really slow you down, unless you have “near Ethernet” speeds
- Be cautious about ENV files, and stuff that loads at startup (e.g. toolbars, bitmaps for buttons) over the VPN

Customizing

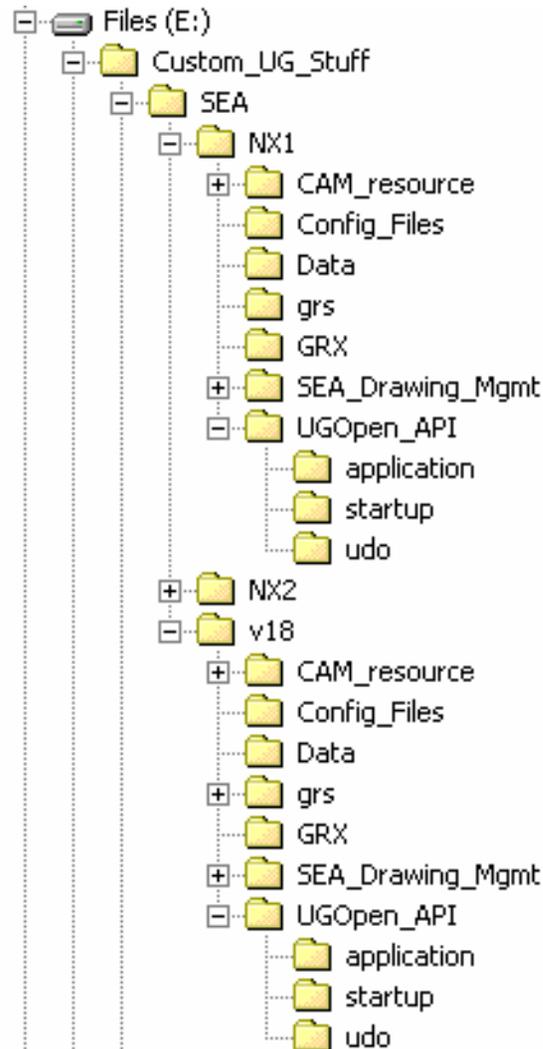
- General
- ENV variables
- ENV file
- Customer Defaults

Customizing – General*

- In General, it is a VERY BAD IDEA to modify any file ***IN*** the NX install tree, as Maintenance Release installs may overwrite your customizations.
- Instead, set up a network share (so ALL users see the same environment), copy needed files there, then start customizing. E.g. (from a client perspective):
 - S:\NXCustom\NX3\Config_Files*.*

Customizing – General (cont)

- My folder structure looks something like:



How much to customize?

- Don't go wild customizing EVERYTHING.
- But don't be timid and customize NOTHING.
- Plusses:
 - If something saves a user 10 seconds, and they do it 20 times a day, it will save them 15 min/week, or 12+ hours a year.
 - Things will be more consistent if they start the way users like
- Minuses
 - The more you customize, the more work you will have to do to upgrade to new releases.
 - If you change menu names too much, GTAC won't be able to help
- So it's a trade-off, and YOU will have to decide the appropriate balance point.
- Let users tell you what annoys them

NX3 and the Registry

- Although NX is currently also using the registry to store some of this data, they are moving away from it, as the registry is not platform independent.

NX3 and the Registry (2)*

- USER configuration (window and toolbar layout, etc.) is stored in an MTX file.
- Stored in:
`C:\Documents and Settings\[User's name]\Local Settings\Application Data\Unigraphics Solutions\NX\user.mtx`
- If a user moves to a different system, copy that file to copy their setup.
- Data is in XML format
 - However, no XSL is supplied
- Supposedly, there is an UNDOCUMENTED ENV to specify this file's location (use at your own risk)
 - UGII_USER_PROFILE_DIR

ENV variables*

- Defined at OS level during install:
 - **UGII_BASE_DIR** (e.g. `E:\Unigraphics NX2`)
 - **UGII_ROOT_DIR** (`%UGII_BASE_DIR%\UGII`)
 - **UGII_LICENSE_FILE** (`27000@YOURFLEXSERVER`)
 - This is changed for redundant/multiple FlexLM servers
- One important ENV to know about for customization (not created by default during install):
 - **UGII_ENV_FILE** (`s:\Custom\NX3\Config_Files\My_Env.Dat`)
 - This points to a file that sets up MANY other ENVs
- The above 4 are the most important – by managing them, you can easily manage multiple users, multiple environments, and concurrent use of multiple NX versions.

Where can I define an ENV?

- OS - System Level [must be administrator]
 - Start ▶ Control Panels ▶ System ▶ “Advanced” Tab ▶ “Environment Variables” Button ▶ bottom half of dialog
 - [MS Win2k resource kit] SetX.Exe NAME VALUE -m
- OS - User Level (same as above, top half of dialog)
 - SetX.Exe (without “-m”)
- Above 2 are “persistent” – still exist if you log off/on
- OS - Dynamically [only in context of a process]
 - e.g. in a .Bat (or other scripting language) procedure
 - `Set NAME=VALUE`
- ENV file (e.g. UGII_Env.Dat)
 - Only available INSIDE of a NX session/program
- (NXOpen [API] call) UF_set_variable()

Which ENV will be used?

- First one found in following list “wins”:
 - Startup.Bat (defined in current process)
 - OS level ENV – “User” ENV
 - OS level ENV – “System” ENV
 - Env.Dat file – First one found (including “#include”d files) overrides subsequent ones
- (NXOpen [API] call) UF_set_variable() overrides others (Not visible in log file or before .DLL was run, so pretty useless except for NX Open programs).

How do I use an ENV variable?

- In a DOS .Bat file, or in Customer Defaults, use
 - `%Variable%`
- In an Env.Dat or *.Def file, use
 - `${variable}`
- I will use both in this presentation (I hope I have them correct 😊)

UGII_Env.Dat (unix: .ugii_env)

- This file manages all other configuration files & many default file locations, using ENV variables
- One Env.Dat file can “#include” another, e.g. in My_Env.Dat I can have a line:

```
#include ${UGII_ROOT_DIR}\ugii_env.dat
```
- You can have multiple “#include”s, or nest them
- When upgrading:
 - Use text differences program (discussed later) to see what changed, and if there are any new ENVs you might want to set up

UGII_Env.Dat (2)

- Recommendation:
 - Create your custom file in a shared area on a server
 - On all workstations:
 - point UGII_ENV_FILE at that shared file.
 - In your custom file, ONLY put the ENVs that you actually CHANGE.
 - Then “#include” the NX default one at the END of your file:

```
#include ${UGII_ROOT_DIR}\ugii_env.dat
```

- This makes upgrading to new versions MUCH easier.
- Remember – FIRST DEFINITION WINS!

UGII_Env.Dat (cont.)*

- You can use one ENV to define another one:

```
CUSTOM_ROOT_DIR=S:\NXCustom\NX3
```

```
UGII_PATDIR=${CUSTOM_ROOT_DIR}\Patterns\
```

- You can add your own ENV variables
 - “CUSTOM_ROOT_DIR” in the above example
- Note: Starting with NX1, there is a “version” variable:

```
${UGII_VERSION}
```

 - See notes page for BBS post with this info
- “#” in 1st char indicates a comment line, except:
 - #include; #if - #else - #endif (and other similar ones)
- “#if”, “#else” and “#endif” (and others) constructs
 - ONLY Documented in UNIX System Admin Guide (!)
 - See notes page for (another) BBS post with this info
 - Or UGSolution articles #001-4972694, #001-0537290

Which UGII_ENV.Dat will be used?

- Will be looked for in this order:
 - 1) The one pointed to by the OS ENV variable **UGII_ENV_FILE**
 - 2) The current working directory
 - 3) The user's "home" directory
(\${HOMEDRIVE}\${HOMEPATH})
 - 4) The directory defined by \${UGII_ROOT_DIR}
- (1) can be any file name
- (2 – 4) must be "UGII_Env.Dat" (case blind)
- NOTE:
 - TC Engineering's Env file will override NX's

Useful UGII_ENV.Dat settings*

- **UGII_PATDIR**
 - Default for drawing borders (patterns)
- **UGII_ENGLISH_THREADS, UGII_METRIC_THREADS**
 - If you want to change tap drill sizes, add thread types, etc.
- **UGII_CAM_RESOURCE_DIR or UGII_CAM_CUSTOM_DIR**
 - If you want to customize CAM (see later)
- **UGII_ANNOTATION_DIR** – Standard drafting note text
- **UGII_SMP_ENABLE** – Enable SMP on 2 CPU systems (more later)
- **UGII_TMP_DIR=\${TMP}** – Where syslogs, etc. created (see notes)

Undocumented ENV Settings*

- Used for testing new features
- You'll only find out about them on the BBS, or from other users or UGS people.
- There is NO “complete” list of them (and probably never will be)
- See notes for one pair for NX2

If an ENV is NOT set

- If an ENV points to a specific file
 - E.g.: `UGII_ENGLISH_THREADS`
- And the ENV is not defined anywhere
- NX will typically use the first file found in the search order listed for the ENV file:
 - 1) The current working directory
 - 2) The user's "home" directory
(`{HOMEDRIVE}{HOMEPATH}`)
 - 3) The directory defined by `{UGII_ROOT_DIR}`

Customer Defaults

- Some settings are redundant with UGII_ENV.Dat (you can set either place)
 - It appears that the ENV variable overrides the customer defaults setting
- MOST (but not all) items you can set in the “Preferences” menu have defaults here.
- Terminology:
 - Default = specified “before” you run NX (change very little, once standardized)
 - Preference = something you change “while” running NX

NX3's "New Look"

- In NX3, UGS is migrating the UG_[English/Metric].Def (and other .Def files) files to a *single* XML file.
- Eventually (NX4 or beyond) most/all .Def files will be incorporated into this scheme (don't know about translators...).
- I *believe* the ENV file will also be (eventually) included (but don't quote me on this...).
- Supposedly, this will allow additions/changes to the defaults in maintenance releases.

Customer Defaults Basics

- 4 levels of where data is (or can be) specified:
 - “Default defaults” - built into NX
 - XML data, but DO NOT EDIT AT THIS LEVEL !!!!!!! !!!!!
 - 3 Optional levels that can be changed by the customer:
 - “Site” or “Enterprise”
 - ENV = `UGII_SITE_DIR` (points to folder)
 - “Group”
 - ENV = `UGII_GROUP_DIR` (points to folder)
 - “User”
 - ENV = `UGII_LOCAL_USER_DEFAULTS` (points to file)
- Plus “Preferences” (interactively set by user in NX)

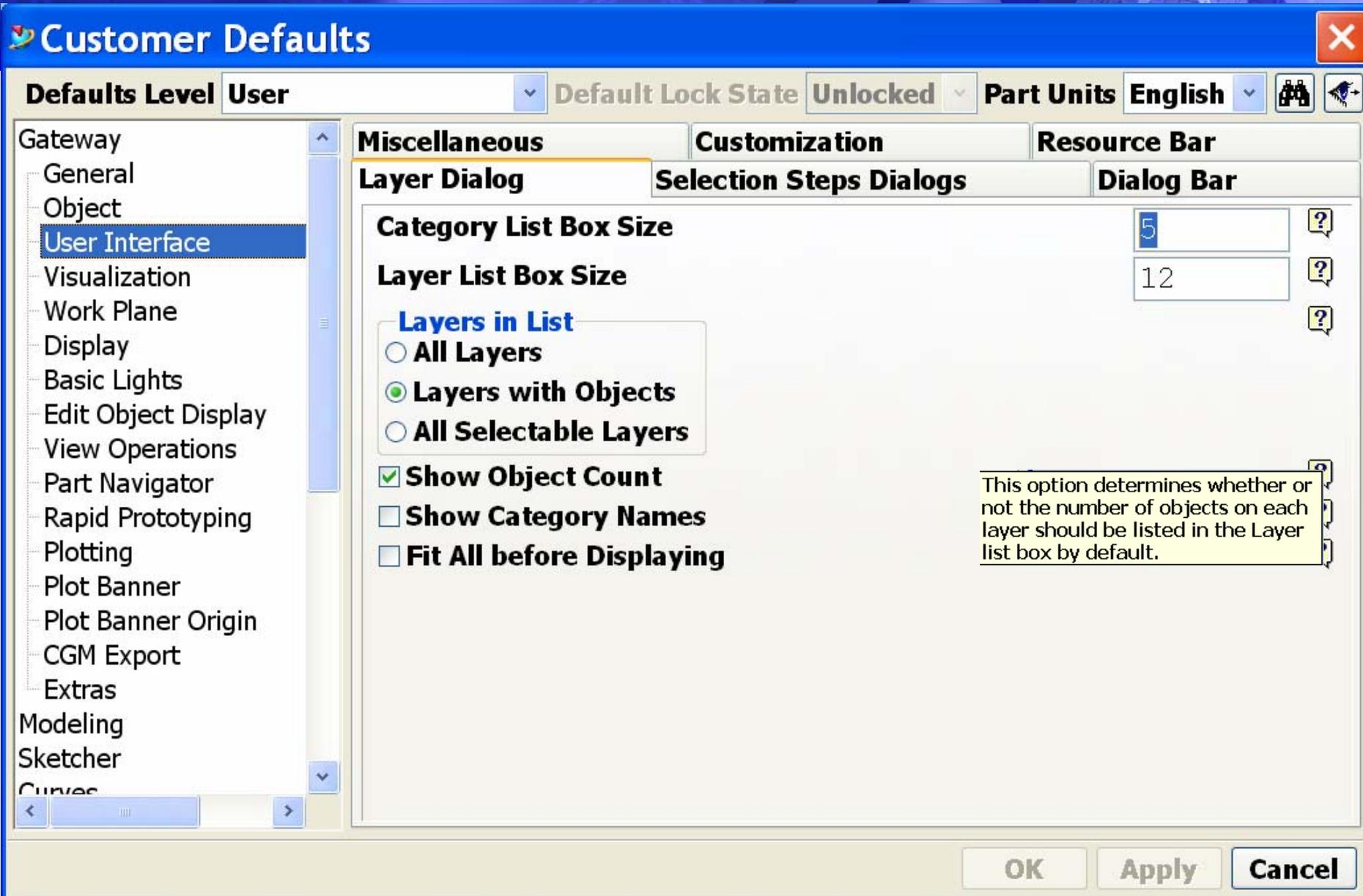
Hierarchy of Defaults

- At the user customizable levels (site/group/user):
 - Data [defaults] can be specified.
 - Data can also be locked from changes at “lower” levels (e.g. if set and locked at site level, groups and users CANNOT change the default).
- Basic procedure when NX starts:
 - “default defaults” come from NX code
 - If site defaults file exists, the defaults are modified per that file.
 - If group defaults file exists, the defaults are modified per that file. (unless locked at site level)
 - If user defaults file exists, the defaults are modified per that file. (unless locked at site or group level)

Defaults Files

- Data stored in XML files. (.DPV + .XSL)
- Only CHANGES are stored at each level in XML data.
- As XSL (XML Style sheet) exists, you can open DPV in Internet Explorer to get a more ‘readable’ version of the items in the file
- Or you can rename the file to “.XML”, open it in Visual Studio (use the “data” view button at the bottom of the VS window)
 - Just rename it back so it is usable by NX!

User Interface



User Interface (cont.)

- Not shown on previous picture:
 - Items ‘locked’ at a higher level will have a little “padlock” icon, also an (optional) user-entered comment.
- Bug in NX2: Locking
 - If both Windows & Unix setting (e.g. default folders)
 - Item will be locked on ONLY the current platform.
 - You must also open default file on OTHER platform & lock there

Customer Defaults - Settings of Interest

- **UG_initialPartDir**
 - Default directory for File ▶ Open
- **UG_bannerDefaultBanner**
 - adds text, at plot time, to plots: date/time plotted, username of who plotted it, etc.
- **UG_initialIgsDir / UG_initialDxfDir / ...**
 - Where to look for IGES/DXF/... files
- **UG_layerListItems/UG_layersInList/UG_showCategoryNames**
 - Default appearance of “layer settings” dialog
- **Drafting_angularUnits**
- **Assemblies_AllowInterPart / Assemblies_AllowPromotions**
- The default colors, line width, & line font of (almost) everything
- Plotting – the default printer group
- **Something like 2000+ settings in total**

Customer Defaults – Setting Types

- Note there are TWO types of defaults:
- Session Defaults
 - Maintained in SESSION.
 - When you exit NX, they revert to defaults
 - E.g. X-Y-Z sizes of block – if changed, while you are IN a session, they will stay at the “new” values when you make the next block. Exit NX & re-enter, they get reset to the default
- Part Specific Defaults
 - Stored IN the part
 - NEW parts get settings from the customer defaults
 - EXISTING parts retain settings from when they were last saved
 - customer defaults are NOT applied (unless you apply them yourself)
 - E.g. the color table

Upgrading Customer Defaults

- There is a whole thread discussing this in the BBS newsgroups (nx.cad group, February 11-15, 2005):
 - Customer Defaults: NX 3 to NX 4
- NX4 implements an “Export all to spreadsheet”
- NX4 will include version that an item was added
 - So new ones will have “NX4” (or whatever)

Other CAD Configuration Files

- thd_english.dat; thd_metric.dat – definitions for symbolic/detailed threads. Edit to:
 - Change tap drills to your shop's default
 - Add thread sizes/classes.
 - You can even (in conjunction with Customer Defaults) add new thread types (e.g. for Helicoil® inserts)

Mixed Site Issues (Unix and MS Windows)

- Assemblies store COMPLETE path for each component.
- Unix paths do NOT look like MS-Windows paths, so components may not be found if assembly was last saved on the “other” OS.
- If you automatically load assembly components using “As Saved” option:
 - Set ENV `UGII_DIRECTORY_MAP_FILE`
 - This maps paths between unix↔Windows formats (e.g. “/usr/ugparts/components” to “S:\Parts\Components”)
- NOT an issue in TC Engineering
 - Path not stored

CAM Customization

- See Docs – Manufacturing ▶ General
 - “Customizing your environment” chapter
- A whole topic by itself, but I’ll cover a couple things here
- Note you can define tools in the tool libraries to have “real” names (“Drill .570”), BUT they MUST be unique real names
- You can customize LOTS of stuff in CAM – read the docs and try things out.
- See some of Jim Maynard’s whitepapers for “out-of-the-box” ideas

CAM Resources

- In the past, UGS recommended (if you were going to customize CAM), to copy the whole CAM resource tree (`%UGII_BASE_DIR%\MACH\resource`) to another location.
 - Then point ENV `UGII_CAM_RESOURCE_DIR` at it
- Now they (and I) are saying to customize it in: `%UGII_BASE_DIR%\MACH\Custom\...`
 - Or point the ENVs (as required) at an equivalent directory tree in another (shared) location.
 - Replicate ONLY the required parts of the folder structure (and files) from `\mach\resource`

CAM Resources - What to do?

- Note that using the new method (with the Custom tree) will take as much advantage as possible of any MR bug fixes.
- Note YOU will have to figure out if any files you have in you custom tree need to be updated because a MR/MP changed something:
 - You need to figure out what files were changed in `%UGII_BASE_DIR%\MACH\resource`
 - Did you make custom versions of any of those files?
 - So [I'm thinking], if you change any files, keep a copy of the original file (unchanged) in your Custom tree, so you can compare it to the one UGS might have changed in a MR/MP install?

Starting NX*

- UGS used to use UGII.Bat to start NX
- Now directly calls one of:
 - Ugraf.Exe
 - UGS_Router.Exe
- But you can still use a .Bat (or other scripting tool)
 - Ugii.Bat is still there if you want to use it
 - Remember those 4 ENVs I talked about earlier?
- .Bat [or other] scripts may be required if you are going to concurrently run multiple versions of NX
 - Using ENV variable `#{UGII_VERSION}` may eliminate this
- Will require editing file types in Explorer (or, equivalently, edit the registry) if you want a .Bat to be used instead of Ugraf.Exe as the default action for .Prt files.

Ugraf/UGS_Router Parameters*

- Each time you run Ugraf, it starts a new NX session
 - Parameters to Ugraf.Exe are in the docs:
 - Gateway ▶ Gateway ▶ Using NX ▶ Starting NX
 - ▶ Windows ▶ Command Line... topic
 - Note the “-retrieve:” option can also specify a bookmark file
- UGS_Router:
 - Added to enable starting NX inside another app (e.g. PDM)
 - Uses an existing session if one is available
 - Otherwise uses NEWEST NX version it can find
 - UNLESS you add version switch:
`ugs_router -ug -version=V18.0`
 - Parameters: See notes page for BBSnotes post

Setting up Plotters*

- I'm not too familiar with the process in NX3
- In NX3, File ▶ Print now uses the same underlying code (SDI) as File ▶ Plot (see BBS post in notes for details)
 - ENV to adjust line widths is gone in NX3 (??)
- Docs (some also on the CD) – 2 PDF files and Plotting FAQ at:
 - <http://support.ugs.com> ▶ Documentation ▶ NX ▶ NX 3
- If your exact plotter/printer model exists, use it
 - Otherwise, it may be easier to start with a GENERIC printer than fiddle with a “close but not correct” one.
- If there are still issues, try using with the INIT files

Setting up Plotters (2)*

- Users can share the same printer setup by defining the ENV variable: `UGII_SDI_SERVER_CFG_DIR`
- Makes a lot of folders
 - `UGII_SDI_BPS_JOBS_DIR` points at top level folder
- See notes page for .Bat to get file “output to file” to a known location
- SDI queues seem to be very sensitive to paper sizes.

Resources (where to get help)*

- GTAC (your friendly 800 number) (or <http://support.ugs.com>)
 - Talk to a human
- UGSolutions [UGAnswer] (<http://uganswer.ugs.com>)
 - Searchable database – PRs, IRs
 - Also indexes BBS newsgroups (although there are gaps?)
- Newsgroups (BBSNotes) (both news and web interfaces)
 - Conferencing system
 - Ask questions of your peers and UGS tech support/developers
 - Many Groups
 - You can only search on “subject” line
 - Note “unigraphics.*” newsgroups recently renamed “nx.*”
- PLMworld (the users group) (www.plmworld.org)
 - Conferencing (requires “citizen” account)
 - Conference CDs – sometimes.

Need WebKey

Need WebKey

Resources(2)*

- Samples
 - See Rob Cohon's posts on BBS
- Other Web Sites (Steve Vickers, etc)*
- support.ugs.com :

The screenshot displays the GTAC website interface. On the left, a yellow sidebar titled "Tech Support" lists various services: GTAC search, call handling, newsgroups, certification, downloads, documentation, GTAC jobs, contact GTAC, GTAC services, license retrieval, and webkey faq. The main content area is titled "GTAC" and features a grid of "GTAC Featured Services" including "Report an Issue", "Status of an Issue", "GTAC Services", "Download & Upload", "Certifications", "Newsgroups", "Licenses & Passwords", and "Documentation". A red arrow points from the "webkey faq" link in the sidebar to the "WebKey Info" text, which is written in large red font and has another red arrow pointing to the "WebKey" logo in the top right corner of the page. The top right corner also includes a "Select a Country:" dropdown menu set to "GTAC World" and a "Do you have a WebKey?" prompt.

UGSolutions Database

- Used to be known as “UGAnswer”
- Search on text in interactive menus
 - Good for American English users 😊
 - Not so good for localized versions ☹️
- And/Or on Dialog/Syslog error messages/status
- Use all lowercase (so it does case-blind search)
 - Mixed Case or UPPERCASE searches will be case-sensitive
- Use complete words

UGSolutions Interface

PLMSolutions Advanced Search

Keyword(s):

Include all Dates sort by Results Per Page

Search Categories: [Default A](#) [None](#) or **click the product label** to turn on the defaults for the product.

Unigraphics NX

nx sfb faq

UG

v18 v17 archive pv ugopen
 sfb faq

I-DEAS

10 NX Series v9 v8 sfb faq
 Techinfo

E-factory

all sfb faq

Parasolid

sfb faq

Newsletter

Newsletter Articles bbs

Solid Edge

v12 v11 archive sfb faq

iMAN

v8 v7 v6 v5 archive itk
 sfb faq

Metaphase/Sherpa

all sfb faq

Vis Products

all sfb faq

E-vis

all sfb faq

Other Products

sfb faq

[Click here for details](#)

Useful System Admin Books

- For writing .Bat procedures:
 - Windows NT Shell Scripting
Tim Hill
Macmillan Technical Publishing
ISBN: 1-57870-047-7
 - Tim also wrote one on Windows Scripting Host
- For Windows System Administration:
 - Mastering Windows 2000 Server (2nd Edition)
Mark Minasi et. al.
Sybex – ISBN: 0-7821-2774-6
 - 2000 +/- pages!!!!!!
 - Now at 4th edition
 - Also “Mastering Windows 2003 server”
 - Also “Mastering Windows XP Professional”

Useful Security Sites

- If you are interested in securing your computers...
- <http://www.nsa.gov/snac>
 - Microsoft
 - OSes: Windows (NT / 2000 / XP / 2003 / etc.)
 - Applications: Office; IE
 - Servers: Exchange; SQL; IIS
 - Sun Solaris 8
 - Mac OS
 - Oracle
 - Cisco Routers
- Others:
 - www.sans.org, www.cert.org, www.nessus.org, www.honeynet.org,
www.tripwire.com, www.microsoft.com/security

Comparing Text Files

- Some tools I have used to compare text files:
 - WinDiff – MS Windows Resource Kit, Visual Studio
 - WinMerge - <http://sourceforge.net/projects/winmerge/>
 - CompareIt! - <http://www.grigsoft.com/>
 - I'm sure there's lots of others out there...
- Also DOS commands:
 - FC
 - Windiff in command line mode

Remote/Shared Desktops

- If you want to see another user's desktop
 - E.g. to show them how to do something
- Note MS graphics option of choice is DirectX
 - UG Graphics are Open GL
 - Hence MS based options may NOT display NX graphics!
- Many options
 - MS – Remote Assistance (XP only)
 - MS - Win 2000 Server
 - VNC & Derivatives (winvnc, tightvnc) – freeware, but slow
 - Teamcenter Appshare – free (<https://appshare.ugs.com>)
 - Sunbelt.com – r admin (not free)
 - Webx (spelling?) free, or corp. license (\$\$)
 - has OpenGL support
 - Altiris – radmin
 - HP website has tool (MS Windows on HP hardware ONLY)

Miscellany

- Accelerators (CTRL, F keys)
 - List available from within NX:
 - Information ▶ Custom Menu Bar ▶ Accelerators
- 64 bit platforms
 - Covered [I hope] in System SIG presentations by UGS.
- If you want to ZIP an entire UG assembly
 - Free utility at <http://www.w-eng.de/>

Resource Bar*

- Browser Home page (or get rid of IE)
 - Customer Defaults – search for
 - UG_browserHomePage
- Get rid of CAST?
 - Un-Set ENV variable UGII_CAST_HOME
- Others: See Docs
 - Path in notes page
- Background Web Page
 - Used to UGS page over internet!
 - See release notes (Caveats ▶ Gateway)
 - Disable:
 - set default `UG_enableWebBackground` to `no`
 - Or replace with your own



Resource Bar (2)*

- You can also add items
- Customize using PAX files
 - Also XML data
 - Preferences ► Palettes
- Fastener libraries:
 - <http://citizen.plmworld.org>
 - Select “UG NX library” in menubar
 - Includes example of customizing a Palette
- Custom Palette information (resource bar) seems to be stored in the registry:
 - [HKEY_CURRENT_USER\Software\Unigraphics Solutions\NX\3.0\General\Palettes\Custom]
 - You can export that key in Regedit & import it to other users so all see the same resource bar



SMP (Multi Threading)

- NX itself is NOT multi-threaded
- The Parasolid kernel IS multi-threaded for:
 - Booleans, Silhouette curves, and Mass Properties
 - NX3+: faceting
- You will get 25-35% boost in first 3
 - I don't know faceting improvement
- Note rebuilding the model due to an expression change may do LOTS of booleans!
- Set ENV variable to turn it on:
 - `UGII_SMP_ENABLE = 1`
 - Only supported on MS-Windows, HP & Sun systems
 - You might want to set this ENV in the OS, as it is machine specific
- Should NOT hurt if enabled on single CPU systems

Temp Directory Files*

- *.Syslog – Log of session. Includes error text/messages/dumps
 - Should be deleted automatically if “no errors”
 - Note: can be turned into a macro (see notes for BBS post)
 - Help ▶ NX Log File
- *.Rol_Bin – Used to implement “undo”
- *.Jnl_Txt – Parasolid journal file (for debugging)
- You may need to periodically clean these files up
 - Do NOT do cleanup while NX is active
 - Make sure all NX sessions are CLOSED/Exited first

- Help ▶ NX Log File
- Now includes all info to get WebKey
 - See earlier slide
- Also lists exact NX version you are running:
 - `NX 3.0.0.21`
- Lists many (but NOT ALL) ENV variables seen by NX session.

Seed Files

- There are SOME settings you just can't change as an administrator (in the customer defaults), or have an NX Open program do.
- If it is important, one way around this is to use seed files – part files with everything set just the way you want.
- Can have several (e.g. model, drafting, CAM)
- If you do, you might want to alter the behavior of the File ► New dialog
 - E.g. use GRIP or an XN Open program for the user to select a file type & enter the new file's path.
- Or put them in the resource bar (see notes)
- TCE: Seed files can be selected in “Create” menu

Patterns

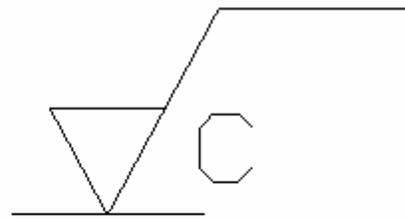
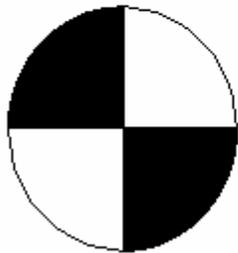
- Typically used for drawing borders
- Regular NX part, but you MUST have “Save Pattern Data” turned on (in File ► Save Options)
- “1 entity” in part where it is used
- “Legacy” way of doing drawing borders
- Requires ENV `UGII_PATDIR` be set.
- Can be combined with NX “Drawing templates”
- UGS recommends:
 - Use patterns for Drawing Borders ONLY
- Notes:
 - Updated EVERY time a part is opened (stored as a link)
 - Can be expanded to edit or remove link
 - Or include Rev level as part of pattern name (INCH_A_Rev0.prt)

Using Custom Symbols for Drawing Borders

- Borders can also be done using Custom Symbols:
 - Can reference attributes
 - So you can set title block text via attributes
 - Stored IN the part file
 - No linked to original
 - Not limited to drawing formats...

User Defined & Custom Symbols

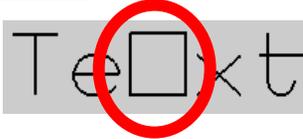
- Way to place common symbols that otherwise aren't supported or found in NX
- Some already supplied, you can add yours
- User Defined = legacy. UGS prefers use of Custom Symbols (that's where emphasis is and enhancements will be implemented)



Load Options

- File ▶ Options ▶ Load Options
- When an assembly is opened, this governs which components are also opened.
- The default setting you want will depend on the size of your assemblies and if/how users work on the sections of an assembly (one user at a time vs. multiple people, each on one part)
- Read and understand the docs
- TC Engineering
 - Dialog is different
 - Functionality is more or less the same

NX Fonts*

- NX fonts are strange beasts, with origins going back 25+ years. Other than normal (7 bit ASCII) characters, there is NO standard for things like which “\$” character is which.
- Some fonts have no lowercase characters.
- The “Full” set is in the International GRIP Library (?)
- See Steve Vicker’s website
- If it’s an issue, I recommend you create a “Font test” part – create a note with all valid regular and “\$” characters. Change its font to see what is available & where.
- You can create your own fonts (see notes for ref).
- Interested in Truetype fonts? See <http://www.w-eng.de/>
- “Box” indicates character is not defined in that font: 
- Defaults to Blockfont if desired font can’t be found

NX Fonts (cont)

- The list of “valid” characters is in the docs [Gateway ▶ Gateway ▶ Gateway Dialog & Menu Reference ▶ File ▶ Utilities ▶ Character Fonts ▶ Edit Font].
 - You can’t use “\$”, except to specify a “\$x” character
 - (\$A, \$a, \$3, \$\$, \$%, etc.)
 - “\$” allows use of an “extended” character set
 - You can’t use “<“ (although “\$<“ is OK)
 - “Control” character for text format/value modification
 - See: Design ▶ Drafting ▶ Appendices ▶ Symbols and Text Control Characters
 - Can be used to include part & object attributes, and expressions in note text
- Fonts are now “binary” – “.FNX” vs. “.FNT”
 - Conversion utility exists to convert both ways

NX Colors*

- Originally, there were 7, then 16.
- V17-NX2 – 216 colors (“Web Standard”), except 1-16 were mapped to the “old” colors
- NX3 – UGS will replace current color tables with new one, colors 1-16 per the “Web Standard”.
- You still can use old color tables, if you wish.
- If you are a new customer, and starting with NX2 or NX3, you might want to use the “new” table now (so you don’t have to change later)
- Parts store their color tables internally, so they won’t look different even if the system tables change.

User Defined Features*

- A way to encapsulate several features in a controlled way, and have them show up as ONE feature in the Model Navigator
- You can control which parameters the user can edit
- 2 Versions (when creating UDFs, not when using them)
 - Customer Defaults - `solids_NewUdf`
 - Pre-v15 – tend to be more robust
 - Post v15 – can have CAM operations added
- May require setup in both ENV and Cust. Defaults
 - Setup changed in NX1 – see notes page for customizing
- Symbolic threads tend to be a pain
- Debugging failures is hard

Part Families*

- Using the spreadsheet, you can create a Family of Parts (e.g. bolts, nuts, and washers)
- One way to get “opposite hand” (mirrored) parts
- You can also create “Assembly” families (e.g. a Bolt + nut + 2 washers)
- Control file name, expressions, attributes, features, etc.
- See the Fasteners library referenced in the resource bar slides for an example
- TC Engineering – Be careful – ownership of members generated “on the fly”
 - See slide notes

Utility Programs

- In \UGII\ folder
 - UGPC.Exe – lists components of parts
 - UG_Convert_Part.Exe – convert inch to metric, or vice-versa
 - Refile_Part.Exe – re-saves part in “current” version (useful for upgrading standard parts)
 - UG_Inspect.Exe – inspect part, also extract Parasolid data.
 - Several more
- Use “-H” argument to get options/arguments
- Most (all?) DO NOT WORK in TC Engineering environment:
 - Ugmanager_refile.bat DOES work

Other Customization Tools

- Macros
- Journaling
- User Tools
- Changing Menu Items (Menuscript et. al.)
- Brief description of automation tools in NX (GRIP, API, ++, KF, etc.)
- Other useful programming/scripting languages

Macros*

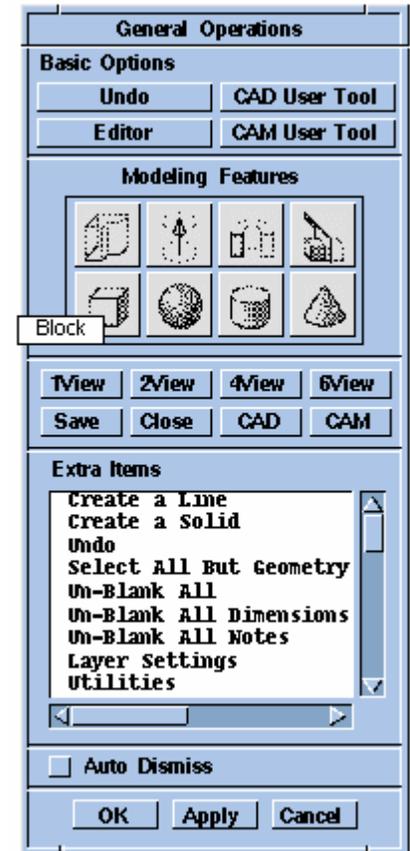
- Record items picked & data entered on menus and dialogs
- NOT guaranteed to work from release to release
- Will capture ENTIRE dialog, unless you edit the macro (I.e. if you change one item in a preferences dialog, the macro will capture and apply ALL the settings in that dialog)
 - Post on BBS from John Baker discusses the “magic numbers”
- Macros NOT nearly as powerful as Ideas program files
- NOTE: Syslog files are (essentially) macros. If you crash (actually a little more complicated than this):
 - Edit the syslog to remove all up to last save, and crash at end
 - Rename to macro & run

Journaling

- New in NX3
- Based on Visual Basic
 - Does NOT support ALL of VB
 - VB is both create & run (replay)
 - You can CREATE a VC++ or Java journal (not run)
- If you are writing an NX Open .Net program, it's a great way to start.
- Does NOT YET cover all of NX
 - Projected 30+% coverage of NX for NX4
- Editable
- MUCH more powerful than macros

User Tools

- Create your own palette/dialog, with buttons & menus to run Macros, GRIP or UGOpen programs, etc.
- Legacy capability, you probably want to use Menuscript and toolbars instead



Altering NX's Menu Behavior*

- User Exits
- Menuscript (more powerful user exit)
- Advanced options (requires setup)
 - Automatic (run when NX starts)
 - UDOs (User Defined Objects)
 - Applications

User Exits

- Legacy
- There are about 30 of these
- You completely replace the NX functionality with your own.
- Typically places where you deal with files
 - File ▶ Open; File ▶ Save; File ▶ New; etc.
- Set up with ENV variables
- You **MUST** write GRIP or NX Open program to implement your functionality

“Standard” Folder Structure*

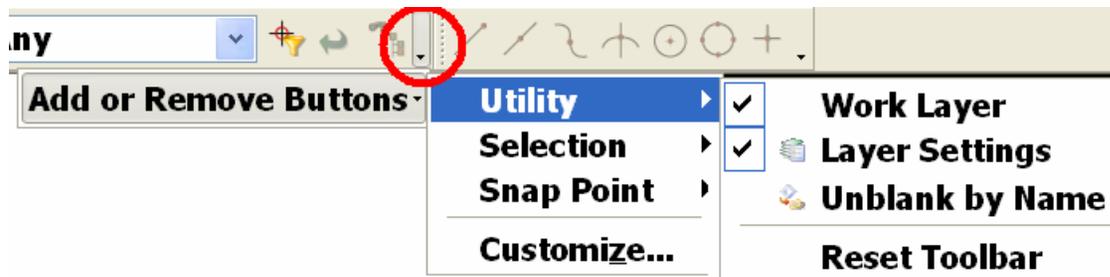
- Create a folder and 1-4 subfolders:
 - `C:\Custom_UG_stuff\NX3\UGOpen_API`
 - This folder will contain 1-4 other sub-folders:
 - `Startup` (required)
 - `Application` (optional)
 - `UDO` (optional)
 - `DFA` (optional)
- Edit `UGII_ENV.Dat`, point `UGII_USER_DIR` (or `UGII_SITE_DIR`) at the folder created above
 - `UGII_USER_DIR= C:\Custom_UG_stuff\NX1\UGOpen_API`
- Alternate method on Notes page (uses `ENV_UGII_CUSTOM_DIRECTORY_FILE`)
- Also maybe set up `UGII_INITIAL_UFUN_DIR`

Menuscript

- Allows COMPLETE customization of NX's menus:
 - Remove items completely
 - Add new items (e.g. "File ▶ Calculator")
 - With NX Open programming:
 - Add one or more programs to be run BEFORE standard NX functionality
 - Add one or more programs to be run AFTER standard NX functionality
 - Completely replace standard NX functionality
- **Used to** require ENV to be set (**not** anymore)
 - `UGII_MENUSCRIPT_SESSION=true`
- UGS .Men files in `..\UGII\menus`

Customizing Toolbars*

- Shared setup pretty much requires use of “standard” folder structure from a couple slides ago.
 - If you add a new toolbar – put in \startup\
 - If you want to change/replace an EXISTING toolbar – put in \Application\
 - See John Baker’s post from the BBS in notes page for more details.
 - Note NX3 has a toolbar editor:



Customizing Toolbars (cont.)

- In conjunction with Menuscript and the automation tools (next slide), this is a powerful way to enhance the user interface

NX Automation Tools*

- GRIP; GRIP/NC – “APT” like, “traditional” language
- NXOpen API (AKA “User Function” or “Ufunc”) – API functions callable from C/C++
- NXOpen++ - Object Oriented interface to NX. C++ Class library of NX classes (e.g. UgLine class)
- Knowledge Fusion (KF/KBE/KDA) – Knowledge (rule) based “programming” tool, based on Intent! from Heide [name has been changed to ???]
- UI/Styler – GUI tool for dialogs (usable in API, ++, and KF)
- New “Common API” – VB/VC#/J#/VC++.NET
 - Java (NX4)

Other useful programming languages*

- tcl – used by CAM group for:
 - UG/Post posts (including those used to create CLS files)
 - Shop Docs output
 - Reading many of the library files (e.g. tools, feed/speed tables, etc.)
 - Process Wizards (?old style)
- Also used for UDF libraries, and elsewhere
- UG/Posts & Shop Docs have much customization added, so the “tcl” is hidden, BUT it’s there, and can be used if you need it.
- TK dialogs can be used, but in a round-about manner
- Learning tcl – see notes page

XML

- While not a “programming language” ...
- A lot of tcl may be replaced by XML in future
- Already used for resource bars, etc.

Thanks...

- John Baker (UGS PLM Solutions)
- Rob Cohon (UGS PLM Solutions)
- Ben Loosli (Ingersoll-Rand Company)
- Stan York (Wright Medical Technology)
- Mark Hoover & Mary Nebrig
- Paul Phillips (Kennametal)

For reviewing and commenting on this presentation
(and/or earlier versions or parts thereof)

Questions?

- Hope you enjoyed this, and got something out of it.
- Suggestions/comments welcome – please let me know what I can do to improve this seminar.
 - Ken Akerboom
 - kakerboom@moog.com
 - (716) 687-7939