October 15, 2007

        Bentley Design++ XM Edition Release Notes
        ==

Welcome to Bentley Design++ XM Edition. This is the first major
Design++ release since Bentley Systems acquired Design Power, Inc. The
main emphasis in this release has been on productizing Design++ as a
true Bentley product.

Design++ XM Edition is supported on Windows 2000/XP/Server 2003
platforms. Even though Windows Vista is not yet officially supported,
preliminary testing indicates that Design++ runs fine also on Vista.

HIGHLIGHTS

    BENTLEY SELECT LICENSING

    Design++ licensing mechanism is reimplemented using Bentley's
    SELECT licensing. As a result, Design++ no longer requires a
    constant file access with the license server. Instead, after the
    initial activation, Design++ XM Edition only needs to connect
    with the license server every 30 days to maintain its
    activation. The 30-day periods between connections can be spent
    off the network without any loss of productivity.

    Once you no longer need access to previous Design++ versions, you
    can uninstall the old CrypKey based license manager. Instructions
    for uninstalling CrypKey are provided later in these release
    notes.

    NEW DOT NOTATION FOR DESIGN RULES INTRODUCED

    The new Dot Notation based component and attribute referencing
    syntax makes design rules shorter and more readable. Here are
    some examples.

    Compact component-attribute referencing:
        Dot Notation   Classic Syntax
        ?component   (:? component)
        ?component.attribute   (:? component attribute)

    Keywords for common access scenarios:
        Dot Notation   Classic Syntax
        ?parent.attribute   (:? (:parent self) attribute)
        ?root.attribute   (:? (dpp-model-root) attribute)

    ENHANCED DESIGN RULE EDITOR

    Design Rule Editor is enhanced to support the new Dot Notation
    syntax extension for Design Rule Language.

    In addition, a new floating Context Sensitive Menu is introduced.
    The floating menu automatically positions itself below the
    current cursor location for easy selection. It also disappears
    when not needed and then reappears as the user continues to edit
    the rule.

    SELF-GUIDED TUTORIAL INCLUDED

    This release comes with a hands-on tutorial for learning
    Design++. The 60-page Vessel Exercise tutorial gives a good
    overview of the Design++ development process and is often used in
    Design++ training classes. Before you do anything else with
    Design++, it is highly recommended that you go through this
    tutorial.

    The Vessel Exercise tutorial and the 2 related documents, a set
    of trainer notes and a PowerPoint presentation, are available in
    the tutorial directory under the main documentation directory.

    ENHANCED INSTALLATION

    Design++ now supports silent Setup through InstallShield's silent
    installation capability. With silent installation, there is no
    need for the user to monitor the installation and provide input

via dialog boxes. A silent Setup runs on its own without any user
intervention.

Also, the normal installation is now customizable through an init
file that can be used to specify default values for various
installation settings. User prompts can also be blocked.

MICROSTATION LINK IS NOW CIM BASED

MicroStation link is reimplemented with the CAD Integration
Manager (CIM). As a result, all supported CAD links are now
CIM-based allowing CIM to be embedded with Design++.

SUPPORTED CAD VERSIONS

Design++ XM Edition integrates with the following CAD versions.

AutoCAD: 2000, 2000i, 2002, 2004, 2005, 2006, 2007, and 2008.
MicroStation: V7, V8 2004 Edition, and V8 XM Edition.
Visio: 2000, 2000 SR1, 2002, 2003, and 2007.

INSTALLATION
----

* Silent Setup is supported through InstallShield's silent
  installation capability. With silent installation, there is no need
  for the user to monitor the setup and provide input via dialog
  boxes. A silent Setup runs on its own without any user intervention.
  See Installation Guide for more detailed

* The normal installation is now customizable through an init file,
  which can be used to specify default values for various installation
  settings. User prompts can also be blocked.

LICENSE MANAGER
----

* Design++ licensing mechanism is reimplemented using Bentley's SELECT
  licensing. As a result, Design++ no longer requires a constant
  license directory access on the license server. Instead, after the
  initial activation, Design++ XM Edition only needs to connect with
  the license server every 30 days to maintain its activation. The
  30-day periods between connections can be spent off the network
  without any loss of productivity.

  Design++ product activation is achieved by connecting to a SELECT
  Server, receiving verification that the SELECT Server Name and
  Activation Key are correct, and successfully posting Usage logs. A
  special Product Activation Wizard will assist users in the initial
  activation process. The Activation Wizard can be run at the end of
  Design++ installation or later from within Design++.

  The product activation process is described in detail in the
  Installation Guide. For more information on SELECT licensing, please
  visit Bentley's website at www.bentley.com

* Once you no longer need access to previous Design++ versions, you
  can uninstall the old CrypKey based license manager including the
  Windows service that it has set up. Note that before you uninstall
  CrypKey make sure that there isn't any other applications on your
  workstation that use CrypKey as a license manager.

  To uninstall CrypKey:

    1. Run dppUpdateLicenseManager.exe from the <d++>\misc\crpk
       directory with the -uninstall option in the Windows Command
       Shell:

         C:\d++60\misc\crpk> dppUpdateLicenseManager.exe -uninstall

    2. Delete all Design++ license files (dpp.*), like dpp.41s,
       dpp.ckn, dpp.dxl, dpp.ent, dpp.key, dpp.rst in the C:/windows
       directory.

DESIGN RULE LANGUAGE
----

* Dot Notation syntax for Design Rule Language introduced. Dot

Notation based component and attribute referencing syntax makes
design rules shorter and more readable. Here are some examples.

    Compact component-attribute referencing:
        Dot Notation   Classic Syntax
        ?component   (:? component)
        ?component.attribute   (:? component attribute)

    Keywords for common access scenarios:
        Dot Notation   Classic Syntax
        ?parent.attribute   (:? (:parent self) attribute)
        ?root.attribute   (:? (dpp-model-root) attribute)

    Keywords for component property access:
        Dot Notation   Classic Syntax
        ?component.PARTS   (:parts component)
        ?component.ALL-PARTS   (:all-parts component)
        ?component.PARENT   (:parent component)
        ?component.NAME   (dpp-component-name
                                     (:? component) :internal)
        ?component.PRETTYNAME   (dpp-component-name
                                     (:? component) :pretty)
        ?component.INDEX   (:index component)
        ?component.CLASS   (:class component)

    Functions can be called on results and calls can be chained:
        Dot Notation   Classic Syntax

        ?component.attribute.func(arg) (func (:? component attribute)
                                         arg)

  Design Rule Editor has been enhanced to support the new Dot Notation
  syntax.

DESIGN RULE EDITOR (DRE)
----

* Context Sensitive Menu (CSM) is modified to support the new Dot
  Notation syntax extension for Design Rule Language.

* Commands have been added to convert rules to/from Dot Notation
  syntax.

* The 'Separate' CSM dialog type is modified to move along with the
  main DRE dialog maintaining its relative location.

* A new default CSM type 'Floating' is introduced.

    - Floating menu is automatically positioned below the current
      cursor location.

    - Floating menu disappears when a non-printing character, like
      moving cursor with a arrow key, is typed. It reappears as soon
      as the mouse is clicked or a printing character is typed. The
      menu disappears also when it has nothing to display.

    - If the floating menu is moved, it will lock its position
      relative to the main DRE dialog and retain it until the menu is
      closed. When the floating menu reappears next time, it reverts
      back to its default floating positioning behavior.

C/API
-----

* Added 'const' qualifiers to some '* char' definitions in portability
  function declarations.

COMAPI
------

* Removed some unnecessar Windows Registry key creations.

CAD INTEGRATION MANAGER (CIM)
----

* CIM is now preloaded into Design++ image.

* The Lisp side of old non-native MDL based MicroStation link is
  partially merged with CIM. Now the Design++ function
  dpp-cad-set-cad-system can be used to switch between any supported
  CAD system as long as the project supports the specified
  CAD. Previously this was not possible as MicroStation was not
  implemented with CIM.

* Fixed a memory leak, where the return value from the primitive
  creation functions was not freed.

MICROSTATION LINK
----

* The Lisp side of old non-native MDL based MicroStation link is
  partially merged with CIM. The function dpp-cad-using-cim-p returns
  t but the dpp-cad-using-ms-nocim-p returns nil.

* Header file msdppdefs.h has been renamed to dppdefs.h and the file
  dppextap.h has been removed. Instead of msdppdefs.h you should
  include dppdefs.h.

* All old MDL dppExtap user function aliases have been removed.

* Free function usage has been modified.

  - A string value needs to be returned from function, which will be
    used as the GEO_REP attribute value.

  - Free functions now need to handle the transformations, color, and
    layer settings. In earlier releases this was done by the MS link
    itself.

  - Instead of dppExtAppSetElmDescr, free function needs to call
    function dppExtapMappingSetComponent to create mapping.

  - The pre-Design++ 6.0 free function registration method, where
    index numbers where used to specify free functions, is no longer
    supported. Now all free functions need to be registered using
    string names, for example:

        static freeProcFunc_t free_func_table[] = {
          {"example_Symbol",   extapCreate_cellWithAttributes},
          {"example_Cylinder", extapCreate_cylinder}
        };

* The case in dppExtapAutoCalcXX function names has been changed to
  dppExtapAutocalcXX

* dppExtapXX functions now have const qualifiers for string arguments
  when applicable.

* MS_MDL path is now set to
  <project>/<msint>/mdl;<d++>/<msinterface>/win32-i86, for example:

  C:\d++\projects\geo_test\msv8xm\mdl\;C:\d++\msv8xm-interface\win32-i86\

  Previously the path included also the following directories.
    <d++>/<ms-interface>/mdl
    <d++>/<ms-interface>/mdl/win32-i86.

* With MicroStation XM Edition the display is forced to update after
  every geometry creation and deletion. This has some negative impact
  on the performance.

* Design++ function dpp-cad-eval-user-message now returns NIL on
  error.

* Design++ MS about box now shows the exact version of MDL used to
  compile the link files.

* Level number handling for MS V8 is fixed.

* AutoDrafter clipping plane problem with MS XM Edition is fixed.

* A preview version of Native MDL link version for XM is available,
  for details see <d++>/msxm-interface/Readme.txt.

AUTOCAD/ARX LINK
----

* AutoLisp free primitive is fixed to handle the GEO_COLOR value
  BYLEVEL as a synonym for BYLAYER.

* Some example free primitive functions in dppExtap are fixed to
  return the handle as a freshly allocated string, as required,
  instead of a pointer to a static string.

MISC
----

* Design++ now supports pathnames with spaces.

=====