ZWCAD 2020 Official

Product Release Notes

ZWSOFT PRODUCT TEAM

ZWSOFT | 2019/6/14

Welcome to ZWCAD 2020!

Dear friends,

We are glad to tell you that the long-awaited ZWCAD 2020 Official is available now! After a long time of devoted preparation and development, and thanks for your valuable feedback for the Open Beta, ZWCAD 2020 finally comes with significant new features and improvements, and notably enhanced efficiency and stability. Now we would like to invite you to take a look at this Official version.

This Release Note mainly introduces the performance of efficiency and stability, new features and improvements, API, new commands and system variables, bug fixes, and limitation and notes in ZWCAD 2020.

Yours sincerely,
ZWSOFT Product Team

June 2019

Contents

ZWCAD 2020 Official Release Notes

VERNUM=2019.5.29(46310)

Overview

ZWCAD 2020 has the following new features and improvements:

New Features	Description
Annotation Scale List Reset	If the drawing includes too many scales, a prompt will pop up to remind users to reset scale list when opening drawing.
PDF Underlay Manager	All attached PDF Underlays can be managed in a dialog box.
Data Extraction	Information and data of different objects can be extracted and exported to a table or a .csv/.xls file.
Lisp Debugger	Lisp programs can be debugged.

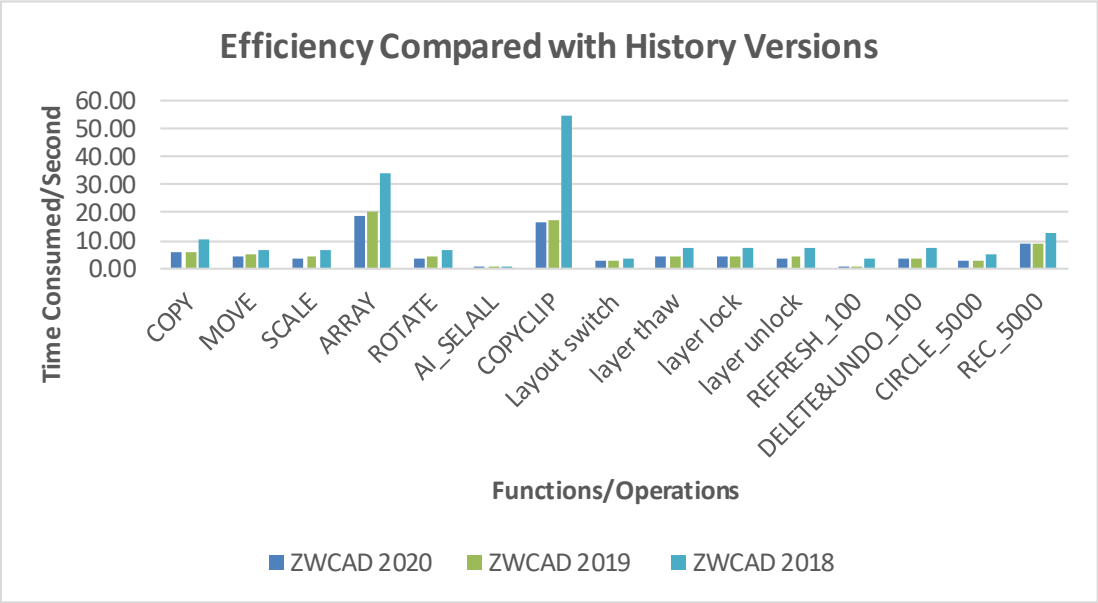
Improvements	Description
Xref Reconstruction	The Xref has been reconstructed to be more powerful.
Color Option in Hatch	The colors of hatch pattern and hatch background can be set in Hatch dialog box.
TK Tracking	It is a new tracking method for users to locate a special point without drawing auxiliary lines.
Selection Set for Overlapped Objects	A dialog box will pop up to help users select overlapped objects.
Smoother Text Explosion	More grips have been added to the 2D polylines after exploding text/mtext, making it more accurate.
Resizable Toolbar	The size of Toolbar can now be adjusted freely.
Block Properties in Properties Panel	Properties of a block are now displayed in the Properties panel in the Block Editor mode.
Find & Replace Text	Users can find and replace letters or words when inputting in the Text box.
More Block Properties in Field	More properties of Dynamic Block have been added to Field.

Efficiency

The following section describes the efficiency tests in this release.

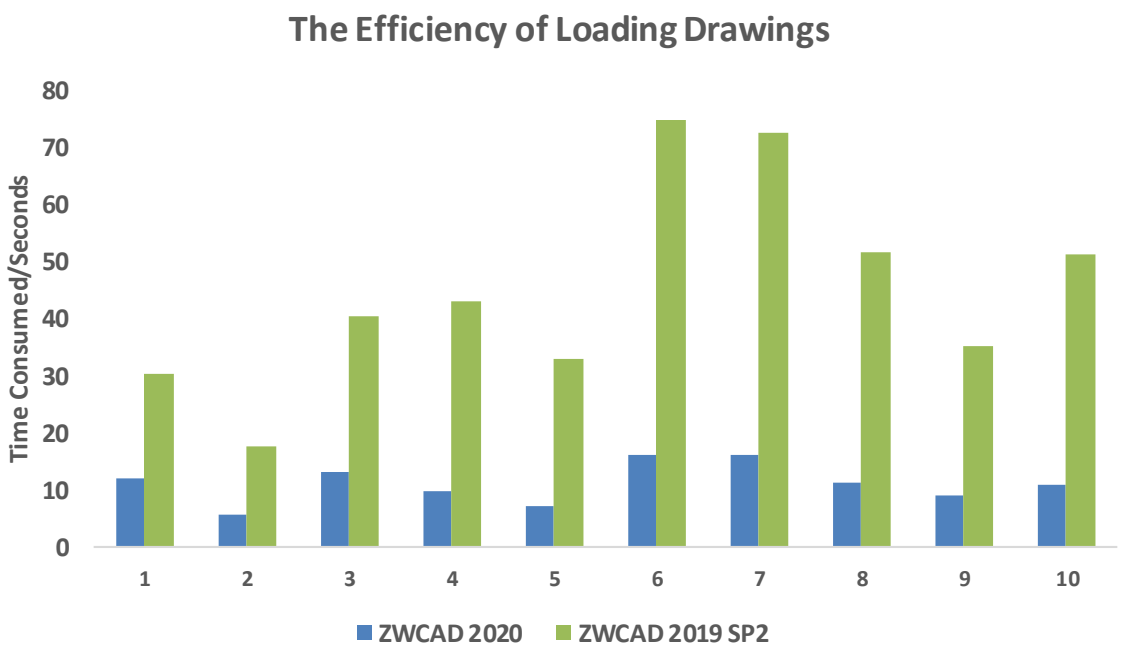
Compare with History Versions

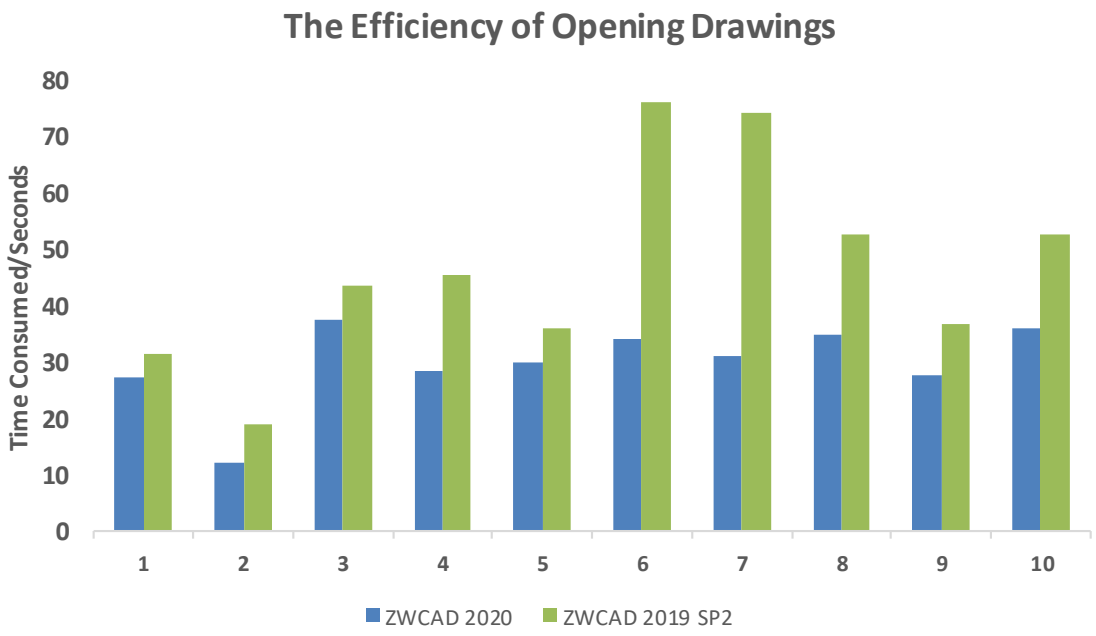
Compared with the previous versions, we can see that basically, the efficiency of the most commonly-used commands and operations in ZWCAD 2020 has been further enhanced, making it the fastest release ever.



Opening & Loading Drawings

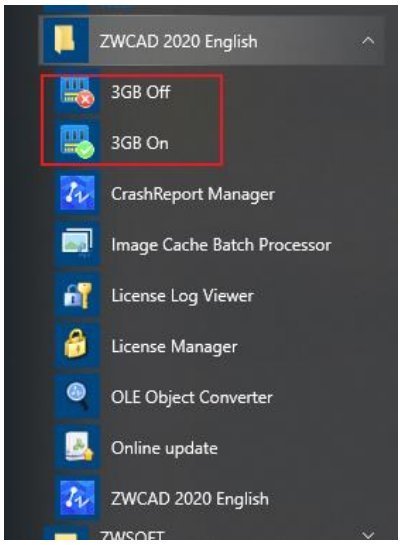
As we all know, the drawings will be loaded before they are opened. After testing 300 typical drawings of different sizes ranging from 4.66 MB to 100.9 MB, the speed of loading drawings has been increased by 73.5%, enabling users to open drawings more quickly, which is 32.4% quicker.


The Efficiency of Loading Drawings
Time Consumed/Seconds
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
ZWCAD 2020
ZWCAD 2019 SP2


The Efficiency of Opening Drawings
Time Consumed/Seconds
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
ZWCAD 2020
ZWCAD 2019 SP2

Large Memory Switch for 32-bit System

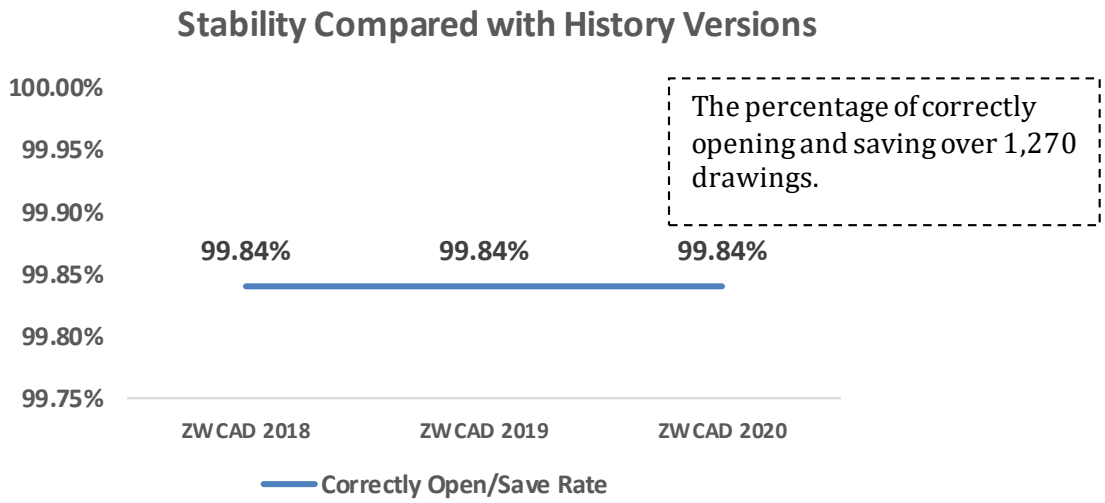
Sometimes when you open large drawings in 32-bit Operation System, crash will happen because the memory is not enough to support the drawing. Now, up to 2.6GB (formerly it is 1.5GB) memory can be released by turning on the Large Memory Switch as below, so that you can open some large drawings with ease.



Stability

The following section describes the stability tests in this release.

The R&D Center has tested over 1,270 drawings, which were extracted from 154,000 comprehensive drawings from the clients, and malfunctioned at least once when conducting Open/Save. The test result was positive by leveling up with ZWCAD 2019 SP2, that is, 99.84% of the drawings could be opened and saved correctly. To summarize, ZWCAD 2020 remains stable.

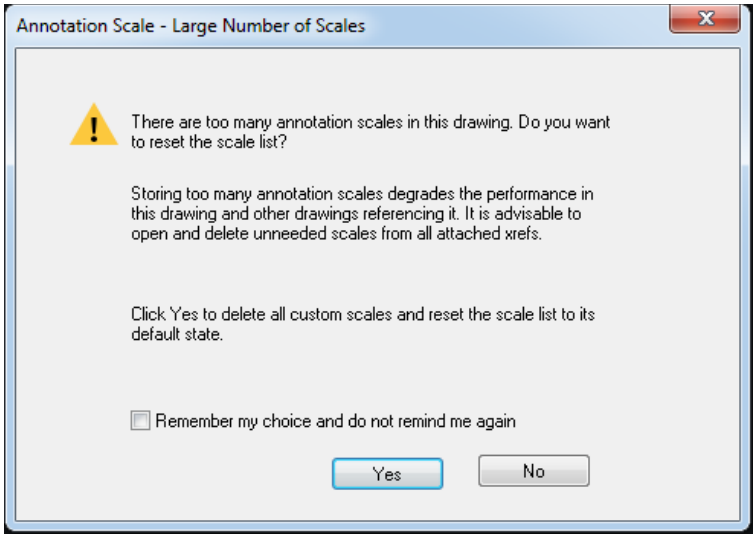


What's more, by systematically solving the crash happening when the trial dialog box pops out during the launch of ZWCAD, the stability of ZWCAD 2020 is predicted to be higher than ZWCAD 2019 SP2.

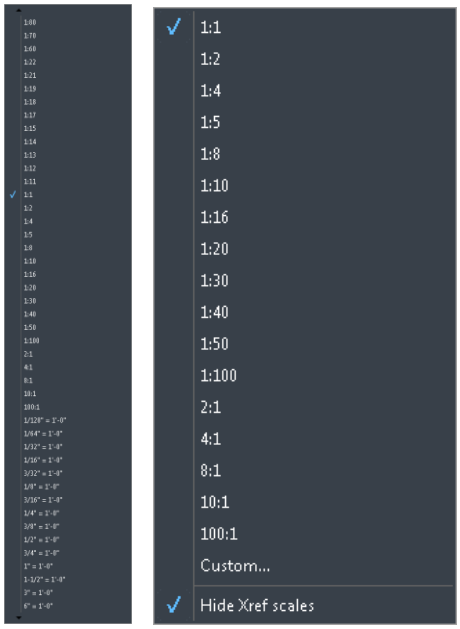
New Features

The following section describes the new features in this release.

Annotation Scale List Reset

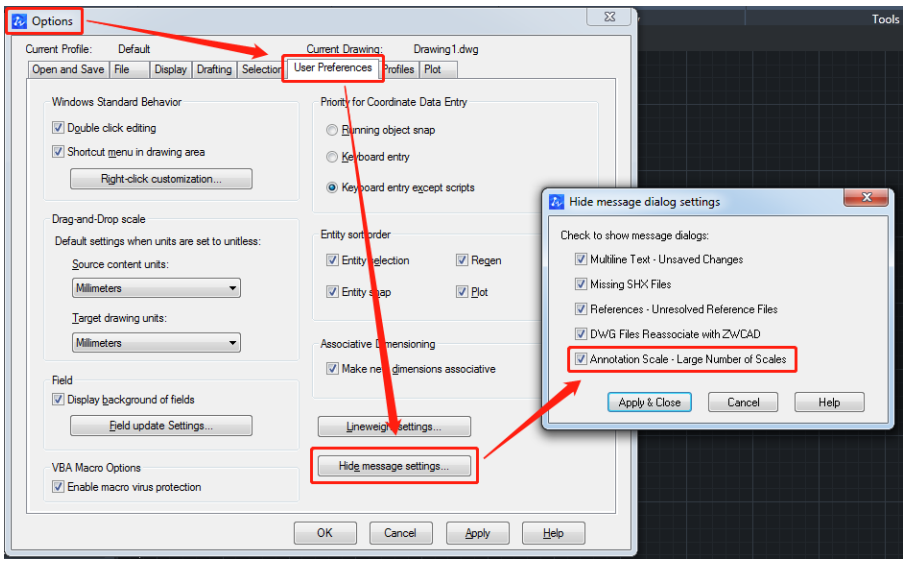


Storing too many annotation scales will spoil the performance of the drawing and its referenced drawings. Now, a prompt will pop up when opening a drawing with lots of annotation scales. It will remind the users to reset scale list to the default state by removing unneeded custom scales, improving the performance when dealing with annotative objects.



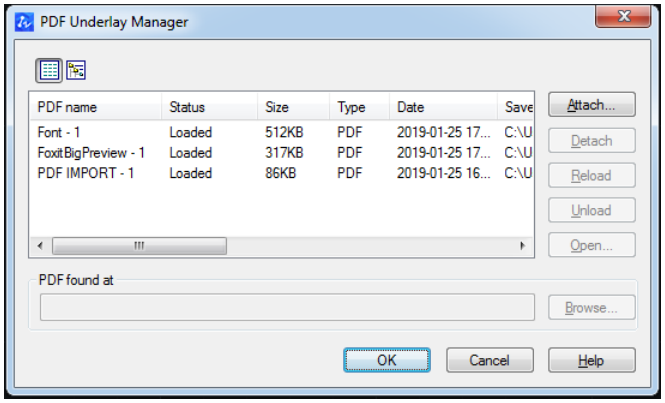
Before and after resetting

If you click *Remember my choice and do not remind me again*, this prompt will never pop up again unless you change the setting in *Option → Users Preferences → Hide message settings → Annotation Scale – Large Number of Scales*.



PDF Underlay Manager

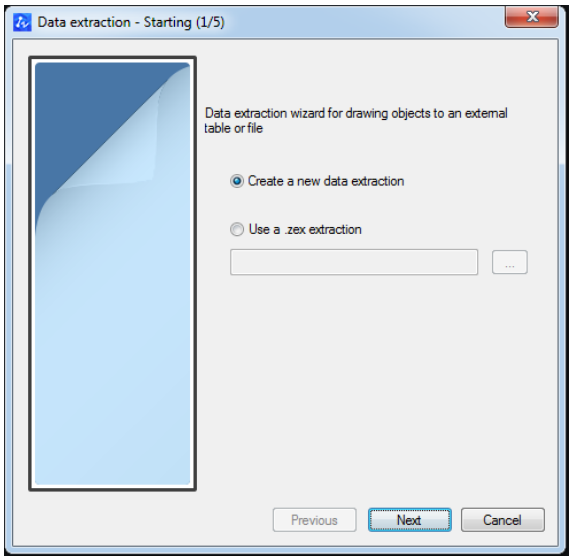
To better manage the attached PDF Underlays, PDF Underlay Manager is now ready for our users in this version. After attaching PDF Underlays, input *PDFATTACH* again, then the dialog box of *PDF Underlay Manager* will pop up for users to check the status, size, type, date, saving path, etc. of all the PDF Underlays.



Users can also attach a new PDF Underlay in this dialog box, and detach, reload, unload or open an existing one, enabling them to manage all the PDF Underlays in the drawing more conveniently.

Data Extraction

Data Extraction enables users to extract various data from objects by filters. There are dozens of different types of data including length, angle, color, layer, area, radius, etc. All these extracted data will be shown in a table or be exported to a file in .csv or .xls format.

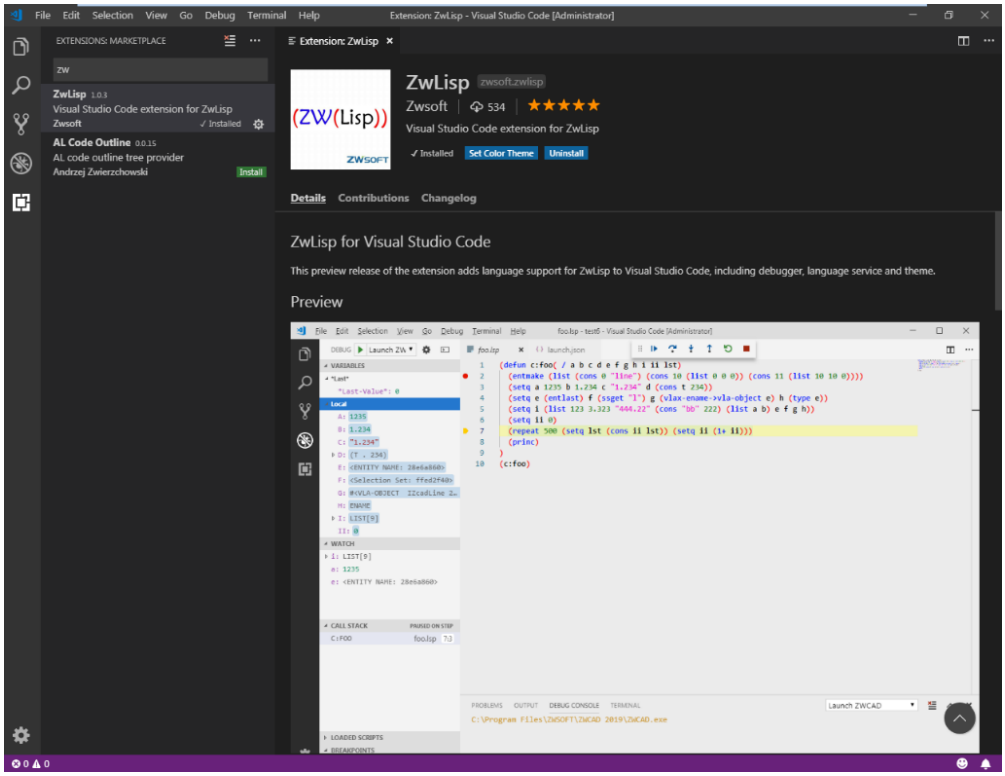


With this feature users can easily acquire the data they need from the drawing. It is especially useful in the situation that numerous data need to be extracted at one time.

Lisp Debugger

Activated by command *VLIDE* or *VLISP*, Lisp Debugger is used to debug lisp programs. Developed based on Visual Studio Code from Microsoft™, it enables developers to debug their lisp programs with breakpoint inspection and modify it easily.

What's more, if users have already installed the Visual Studio Code, they can directly download ZwLisp in Microsoft Extensions Marketplace.


File Edit Selection View Go Debug Terminal Help
Extension: ZwLisp - Visual Studio Code [Administrator]
EXTENSIONS: MARKETPLACE
zw
ZwLisp 1.0.3
Visual Studio Code extension for ZwLisp
Zwsoft
✓ Installed
AL Code Outline 0.0.15
AL code outline tree provider
Andrzej Zwierzchowski
Install
Extension: ZwLisp
(ZW(Lisp))
ZWSOFT
ZwLisp
zwsoft.zwlisp
Zwsoft
534
Visual Studio Code extension for ZwLisp
✓ Installed
Set Color Theme
Uninstall
Details
Contributions
Changelog
ZwLisp for Visual Studio Code
This preview release of the extension adds language support for ZwLisp to Visual Studio Code, including debugger, language service and theme.
Preview

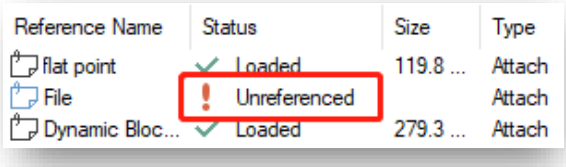
Improvements

The following section describes the improvements in this release.

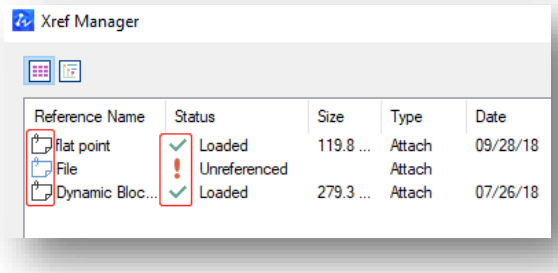
Xref Reconstruction

Here are some improvements thanks to this reconstruction, making the Xref more powerful.

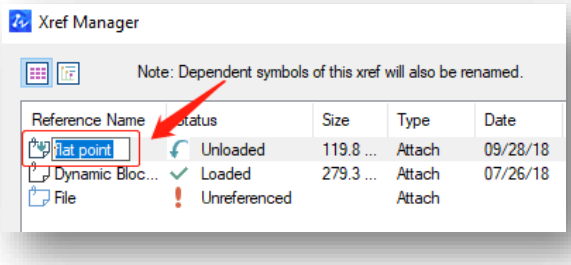
1. “Unreferenced” status can be displayed now.



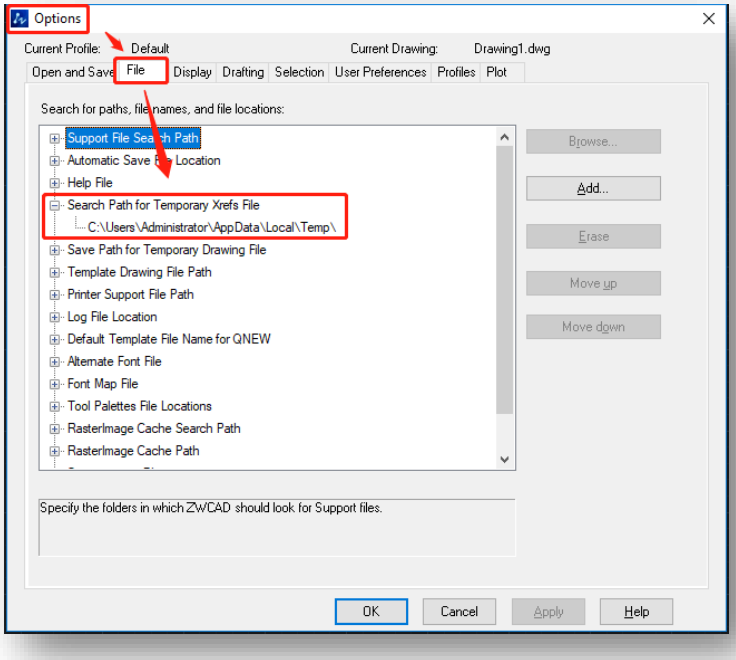
2. New icons have been added to “Reference Name” and “Status”, to better identify the status of files.



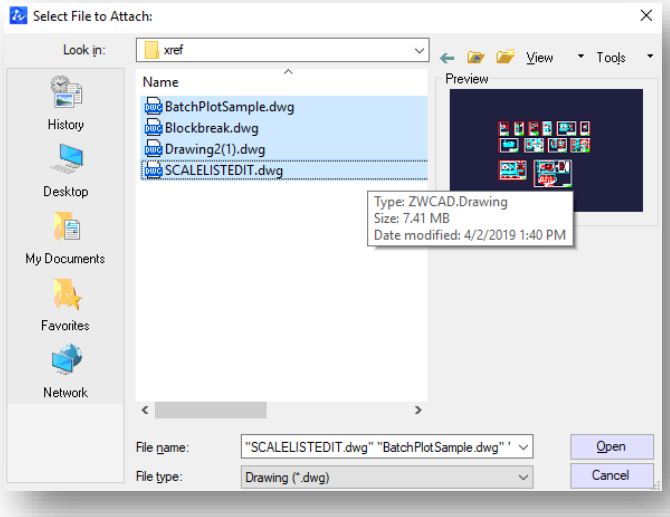
3. Referenced files can be renamed now. However, only the names of the referenced files in the current drawing will be changed, and their original names remains unchanged.



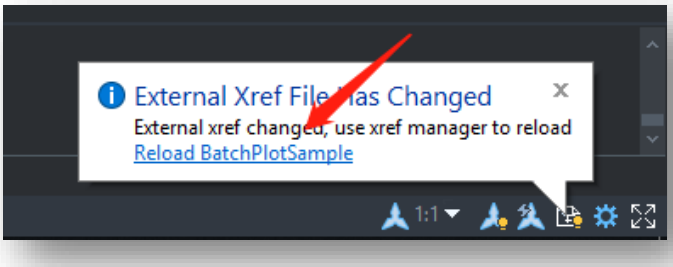
4. “Additional Xref Search Paths”, which can be found in “Options” → “File”, has been removed, making the dialog box simpler.



5. Multiple .dwg drawings can be imported as referenced files at a time when attaching xref.

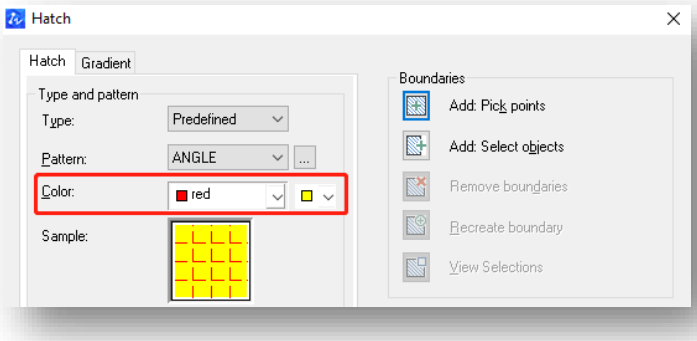


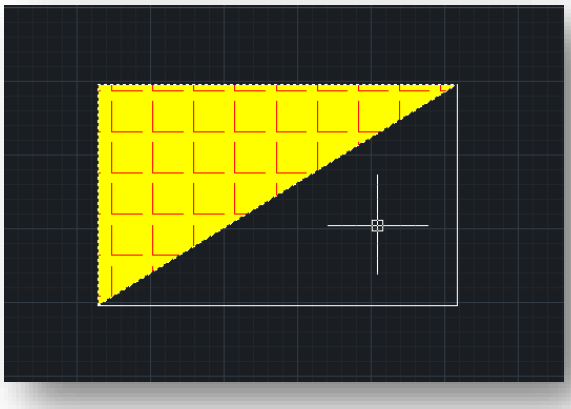
6. When a referenced file is modified, a hint will pop up at the bottom right corner in the current drawing, reminding you to update the referenced file. Users can click the hint to update the referenced file directly without entering the Xref manager.



Color Option in Hatch

"Color" option has been added to the dialog box of Hatch, which means that users can select the colors of hatch pattern and hatch background, and preview the result directly.





TK Tracking

Previously, if the users want to locate a special point, they need to draw some auxiliary lines. Now, with TK Tracking, they can locate it with a virtual auxiliary line. Thus, drawing efficiency has been greatly improved.

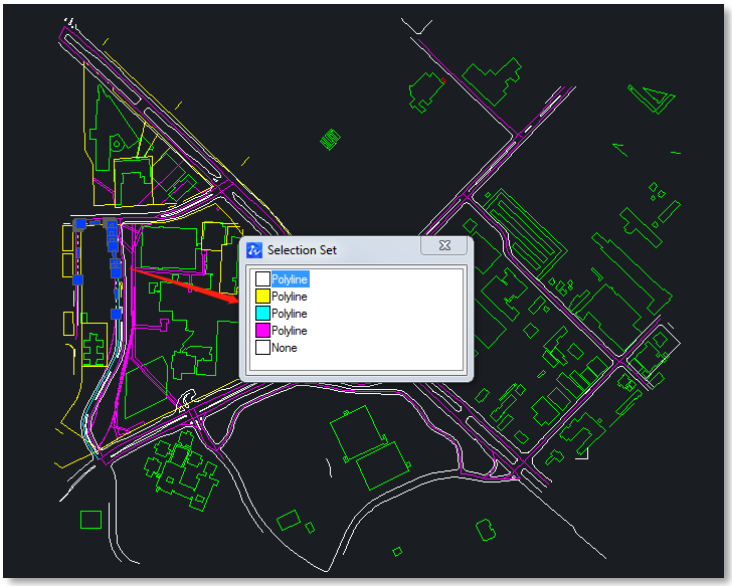
It should be noted that TK Tracking is a kind of in-command object tracking, which means that it can only be used in the middle of drawing or an editing command. For example, you can only use the TK Tracking after activating the Circle command.

```
CIRCLE
Specify the center point of circle or [3P/2P/Ttr (tangency tangency radius)]: tk

First tracking point:
Next point (press ENTER to end tracking):
Next point (press ENTER to end tracking):

Specify circle radius or [Diameter] <120.5215>:70
```

Selection Set for Overlapped Objects

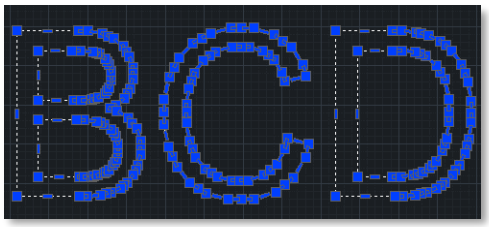
A dialog box called *Selection Set* will pop out when users click the overlapped objects. The color and type of each overlapped object will be displayed in that dialog box, so that users can easily and quickly select the one they want.

It is particularly useful for some complicated drawings, where dozens of objects are overlapped with each other.

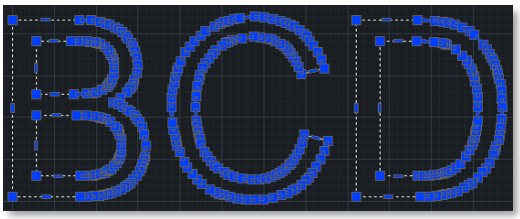


Smoother Text Explosion

By command *TXTEXP*, we can explode text or mtext into 2D polylines with multiple grips. Now more grips have been added, making the 2D polylines smoother and improving the accuracy of text explosion. It will benefit the users who want a more precise laser cutting for text.



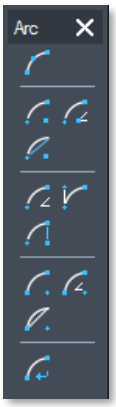
ZWCAD 2019



ZWCAD 2020

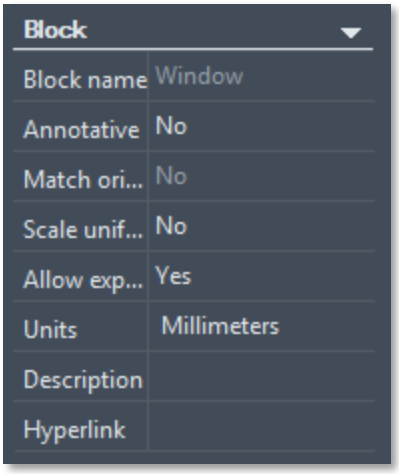
Resizable Toolbar

The Toolbar can be resized now. Users can adjust it to different sizes, rows and shapes according to their habits.



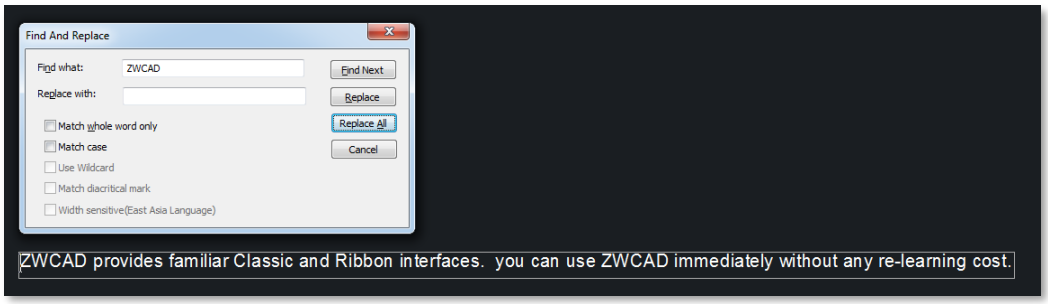
Block Properties in Properties Panel

When operating in the Block Editor, relevant properties of the block can now be displayed in the Properties palette. Users can easily modify the block properties in it, such as Block name, Annotative scale, Match orientation, Scale uniformly, Allow exploding and Units, and can also add Description and Hyperlink.



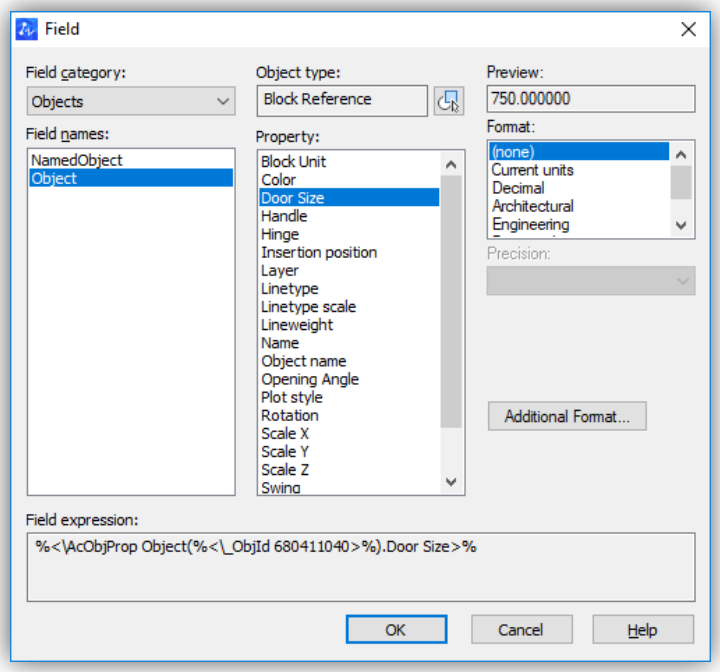
Find & Replace Dialog in Texting

When inputting in the text box, users can find and replace letters or words in the *Find And Replace* dialog box, which appears by pressing *Ctrl + R*.



More Block Properties in Field

More properties of Dynamic Block have been added to Field, giving users more choices to associate with fields and thus making their design more efficient.


Field
Field category:
Objects
Field names:
NamedObject
Object
Object type:
Block Reference
Property:
Block Unit
Color
Door Size
Handle
Hinge
Insertion position
Layer
Linetype
Linetype scale
Lineweight
Name
Object name
Opening Angle
Plot style
Rotation
Scale X
Scale Y
Scale Z
Swing
Preview:
750.000000
Format:
(none)
Current units
Decimal
Architectural
Engineering
Precision:
Additional Format...
Field expression:
%<\AcObjProp Object(%<_ObjId 680411040>%).Door Size>%
OK
Cancel
Help

New Commands & System Variables

The following section describes the new commands and system variables in this release.

New Commands	Description
ALIGNSPACE	Adjust the scale and location of drawings in Layout viewport according to the alignment points which have been specified in Model space and Paper space.
VLIDE	Turn on Lisp Debugger
VLISP	Turn on Lisp Debugger, the same as VLIDE
3DCORBIT	Continuously rotate the views in the 3D working space
DATAEXTRACTION	Extract various properties and data of an object, and insert them to the current drawing as a table, or export them to a .csv / .xls file
DATAEXTRACTIONUPDATE	Update the extracted properties and data according to the actual changes in the drawing

New System Variables	Description
INPUTSEARCHOPTIONFLAGS	Control whether to display a list of commands with similar names while inputting the first few letters in the command line.
SELECTIONCYCLING	Control whether to display the *Selection Set* dialog box for the overlapping objects.
LAYOUTTAB	Control whether to show or hide the Layout tab.
TEXTEDITMODE	Control whether the "DDEDIT" command is used to repeat automatically or edit a single text object
LOADDWGCONCURRENTLYIFPOSSIBLE	Control whether to load .dwg drawings concurrently
TASKBAR	Control whether multiple drawings are displayed as a group or separately in the Windows taskbar.

API

The following section describes the condition of API in this release.

ZRX

ZRX programs running correctly on ZWCAD 2018 Official/SP1/SP2 and ZWCAD 2019 Official/SP1/SP2 can be loaded on ZWCAD 2020 Official directly. 25 were added (highlighted by blue) and 58 were fixed as below.

Bug ID	Modification	Interface
16406	Add	static AcRxClass * AcGiGenericTexture::desc()
10933; 16004; 16705	Add	virtual Acad::ErrorStatus AcDbMPolygon::balanceTree();
10933; 15997	Add	virtual Acad::ErrorStatus AcDbMPolygon::getMPolygonTree(AcDbMPolygonNode *& loopNode) const;
15723	Add	ACDB_PORT AcDbSection::AcDbSection(const AcGePoint3dArray& pts, const AcGeVector3d& verticalDir);
15724	Add	Acad::ErrorStatus acedRestorePreviousUCS();
15829	Add	Acad::ErrorStatus AcDbSection::addVertex(int nInsertAt, const AcGePoint3d& pt);
15829	Add	Acad::ErrorStatus AcDbSection::setState(AcDbSection::State nState);
15829	Add	Acad::ErrorStatus AcDbSection::setVerticalDirection(const AcGeVector3d& dir);
16106	Add	AcGsScreenShot * acgsGetScreenShot(int viewportNumber)
16229	Add	virtual Acad::ErrorStatus AcDb3dSolid::setRecordHistory(bool bRecord);

16229	Add	virtual bool AcDb3dSolid::recordHistory() const;
16401	Add	bool acedLoadPartialMenu(const TCHAR* pszMenuFile);
16424	Add	virtual void AcGsView::zoomExtents(const AcGePoint3d& minPoint, const AcGePoint3d& maxPoint) = 0;
16406	Add	static AcRxClass * AcDbLight::desc()
16406	Add	static AcRxClass * AcGiProceduralTexture::desc()
16406	Add	static AcRxClass * AcGiWoodTexture::desc()
16406	Add	static AcRxClass * AcGiMarbleTexture::desc()
16428	Add	static AcRxClass * AcGiMaterialMap::desc()
16558	Add	static AcRxClass * AcGiShadowParameters::desc()
16561	Add	static AcRxClass * AcDbSun::desc()
16695	Add	static AcRxClass * AcDbMPolygon::desc()
15679	Add	static AcRxClass *AcGiImageFileTexture::desc();
16040	Add	virtual Acad::ErrorStatus AcDb3dSolid::createExtrudedSolid(AcDbEntity* pSweepEnt, const AcGeVector3d& directionVec, AcDbSweepOptions& sweepOptions);
16040	Add	virtual Acad::ErrorStatus AcDb3dSolid::createExtrudedSolid(AcDbEntity* pSweepEnt, const AcDbSubentId& faceSubentId, double height, AcDbSweepOptions& sweepOptions);

16040	Add	virtual Acad::ErrorStatus AcDb3dSolid::createExtrudedSolid(AcDbEntity* pSweepEnt, const AcDbSubentId& faceSubentId, const AcGeVector3d& directionVec, AcDbSweepOptions& sweepOptions);
15732	Fix	int acedSSGet(const char * str, const void * pt1, const void * pt2, const struct resbuf * filter, ads_name ss);
16545	Fix	virtual void AcGeLine3d::getSamplePoints (int numSample, ZcGePoint3dArray& pointArray, ZcGeDoubleArray* pParamArray = NULL) const;
15995	Fix	virtual Acad::ErrorStatus insertMPolygonLoopAt(int loopIndex, const AcGePoint2dArray& vertices, const AcGeDoubleArray& bulges, bool excludeCrossing = true, double tol = AcDbMPolygonCrossingFuzz);
15998	Fix	virtual int numMPolygonLoops() const;
15999	Fix	virtual Acad::ErrorStatus AcDbMPolygon::getMPolygonLoopAt(int loopIndex, AcGePoint2dArray& vertices, AcGeDoubleArray& bulges) const;
16000	Fix	virtual Acad::ErrorStatus AcDbMPolygon::appendMPolygonLoop(const AcGePoint2dArray& vertices, const AcGeDoubleArray& bulges, bool excludeCrossing = true, double tol = AcDbMPolygonCrossingFuzz);
16001	Fix	virtual Acad::ErrorStatus AcDbMPolygon::removeMPolygonLoopAt(int loopIndex);
16002	Fix	virtual void AcDbMPolygon::deleteMPolygonTree(AcDbMPolygonNode* loopNode) const;

16014	Fix	virtual Acad::ErrorStatus AcDbMPolygon::getLoopAtGsMarker(int gsMark, int & loop) const;
16015	Fix	virtual void AcDbMPolygon::getChildLoops(int curLoop, AcGeIntArray& selectedLoopIndexes) const;
16016	Fix	virtual int AcDbMPolygon::getParentLoop(int curLoop) const;
16227	Fix	virtual AcGiViewportGeometry & geometry() const = 0;
15952	Fix	virtual void AcEditorReactor::beginDoubleClick(const AcGePoint3d& clickPoint);
15633; 15634	Fix	ACDBCORE2D_PORT ADESK_SEALED_VIRTUAL Acad::ErrorStatus intersectWith(const AcDbEntity* pEnt, AcDb::Intersect intType, const AcGePlane& projPlane, AcGePoint3dArray& points, Adesk::GsMarker thisGsMarker = 0, Adesk::GsMarker otherGsMarker = 0) const
16017	Fix	int acdbEntMod(const struct resbuf* ent)
16024	Fix	virtual ACMPOLYGON_PORT int AcDbMPolygon::getClosestLoopTo(const AcGePoint3d& worldPt) const
16025	Fix	virtual int AcDbMPolygon::isPointInsideMPolygon(const AcGePoint3d& worldPt, AcGeIntArray& loopsArray, double tol = AcDbMPolygonCrossingFuzz) const
16026	Fix	virtual bool AcDbMPolygon::isPointOnLoopBoundary(const AcGePoint3d& worldPt, int loop, double tol = AcDbMPolygonCrossingFuzz) const

16027	Fix	virtual double AcDbMPolygon::getArea() const
16028	Fix	virtual double AcDbMPolygon::getPerimeter() const
16029	Fix	virtual void AcDbMPolygon::list() const
16035	Fix	Acad::ErrorStatus AcDbEntity::cancel()
16079	Fix	virtual Acad::ErrorStatus AcDbMPolygon::getLoopDirection(int lindex, AcDbMPolygon::loopDir& dir) const;
16080	Fix	virtual Acad::ErrorStatus AcDbMPolygon::setLoopDirection(int lindex, AcDbMPolygon::loopDir dir)
16081	Fix	virtual Acad::ErrorStatus AcDbMPolygon::loopCrossesItself(bool& crosses, bool findAll, AcDbMPolygonCrossingArray& crossingsArray, const AcGePoint2dArray& vertexPts, const AcGeDoubleArray& vertexBulges, double tol = AcDbMPolygonCrossingFuzz) const
16082	Fix	virtual bool AcDbMPolygon::loopCrossesItself(const AcGePoint2dArray& vertexPts, const AcGeDoubleArray& vertexBulges, double tol = AcDbMPolygonCrossingFuzz) const
16083	Fix	virtual Acad::ErrorStatus AcDbMPolygon::loopCrossesMPolygon(bool& crosses, bool findAll, AcDbMPolygonCrossingArray& crossingsArray, const AcGePoint2dArray& testVertexPts, const AcGeDoubleArray& testVertexBulges, double tol = AcDbMPolygonCrossingFuzz) const

16084	Fix	virtual bool AcDbMPolygon::selfCrosses(const AcGePoint2dArray& vertexPts, const AcGeDoubleArray& vertexBulges, double tol = AcDbMPolygonCrossingFuzz) const
16085	Fix	virtual bool AcDbMPolygon::includesTouchingLoops(double tol = AcDbMPolygonCrossingFuzz) const
16095	Fix	void * acrxGetServiceSymbolAddr(const ACHAR * serviceName, const ACHAR* symbol)
16146	Fix	virtual Acad::ErrorStatus AcDbMPolygon::balanceDisplay()
16151	Fix	virtual Acad::ErrorStatus AcDbMPolygon::getArea(double& area)const
16155	Fix	virtual bool AcDbMPolygon::isBalanced() const
16337	Fix	Acad::ErrorStatus wblockCloneObjects(AcDbObjectIdArray& objectIds, AcDbObjectId& owner, AcDbIdMapping& idMap, AcDb::DuplicateRecordCloning drc, bool deferXlation = false)
16603	Fix	virtual Acad::ErrorStatus AcDb3dSolid::getArea(double& area) const;
16603	Fix	virtual Acad::ErrorStatus AcDb3dSolid::getMassProp(double& volume, AcGePoint3d& centroid, double momInertia[3], double prodInertia[3], double prinMoments[3], AcGeVector3d prinAxes[3], double radiiGyration[3], AcDbExtents& extents) const

16161	Fix	virtual HRESULT STDMETHODCALLTYPE GetCategoryName(int CatID, LCID lCid, BSTR __RPC_FAR * pCategoryName) = 0
16918	Fix	Acad::ErrorStatus acdbSaveAs2004(AcDbDatabase* pDb, const ACHAR* fileName);
16278	Fix	virtual Acad::ErrorStatus findFile(ACHAR * pcFullPathOut, int nBufferLength, const ACHAR * pcFilename, AcDbDatabase * pDb = NULL, AcDbHostApplicationServices::FindFileHint hint =kDefault) = 0;
16292	Fix	Acad::ErrorStatus AcDbEllipse::setEndAngle(double ang);
16292	Fix	Acad::ErrorStatus AcDbEllipse::setStartAngle(double ang);
16292	Fix	Acad::ErrorStatus AcDbEllipse::setEndParam(double endParam);
16380	Fix	Acad::ErrorStatus AcDbHatch::getHatchPattern(AcDbSectionSettings::SectionType nSecType, AcDbSectionSettings::Geometry nGeometry, AcDbHatch::HatchPatternType& nPatternType, const ACHAR*& pszPatternName) const;
16382	Fix	inline virtual void AcEdInputContextReactor::beginQuiescentState();
16389	Fix	Acad::ErrorStatus AcDbHatch::setPattern(AcDbHatch::HatchPatternType patType, const ACHAR* patName);
16389	Fix	Acad::ErrorStatus AcDbHatch::setPatternDouble(bool isDouble);

16422	Fix	AcDbHatch::worldDraw() const;
16500	Fix	virtual void CAdUiBitmapButton::DrawButton(CDC& dc, int w, int h, BOOL isDefault, BOOL isDisabled, BOOL isSelected, BOOL isFocused);
16510	Fix	virtual Acad::ErrorStatus AcDbCustomOsnapManager::deactivateOsnapMode(const ACHAR * modeStr) = 0;
16508	Fix	BOOL CAdUiPaletteSet::SetActivePalette(CAdUiPalette* pPalette);
16546	Fix	virtual Acad::ErrorStatus AcDbPlotSettingsValidator::setPlotViewName(AcDbPlotSettings* pPlotSet, const ACHAR * viewName) = 0;
16664	Fix	bool AcadSelectedEntInfo::getSelFenceResultInfo(SelFenceIntersectInfo& selFenceInfo) const
16692	Fix	Acad::ErrorStatus AcDbHatch::evaluateHatch(bool bUnderestimateNumLines = false);
16715	Fix	virtual Acad::ErrorStatus AcDbBlockReference::explodeToOwnerSpace() const;
16899	Fix	void acedCoordFromPixelToWorld(const CPoint & ptIn, acedDwgPoint ptOut)
16768	Fix	AcDbObjectIterator * vertexIterator() const;

16768	Fix	AcDbObjectIterator * attributeIterator() const;
16768	Fix	Acad::ErrorStatus newIterator(AcDbBlockTableRecordIterator*& pIterator, bool atBeginning = true, bool skipDeleted = true) const;

.NET

4 were added (highlighted by blue) and 6 were fixed as below.

Bug ID	Modification	Interface
15842	Add	ZwSoft.ZwCAD.DatabaseServices. SweepOptions.SweepOptions() Constructor
15842	Add	ZwSoft.ZwCAD.DatabaseServices. SweepOptions.SweepOptions (SweepOptions opts) Constructor
15500	Add	ZwSoft.ZwCAD.Windows.Data.HatchPatterns.Instance.AllPatterns Method
10521	Add	ZwSoft.ZwCAD.PlottingServices PlotConfigManager.Devices Property
12389	Fix	ZwSoft.ZwCAD.DatabaseServices Entity.GeometricExtents Property
15897	Fix	ZwSoft.ZwCAD.Windows.PaletteSet.Dock Property
16191	Fix	ZwSoft.ZwCAD.DatabaseServices.Database.CloseInput

16201	Fix	ZwSoft.ZwCAD.DatabaseServices.TransactionManager.GetObject
16247	Fix	ZwSoft.ZwCAD.EditorInput.Editor.SelectWindowPolygon
16255	Fix	ZwSoft.ZwCAD.ApplicationServices.Application.Version

VBA

0 were added and 9 were fixed as below.

Bug ID	Modification	Interface
15857	Fix	PopupMenuItem. Delete Method
15864; 16630	Fix	ModelSpace. AddRegion Method
10867	Fix	object.SendCommand Method
12762	Fix	PopupMenuItem. Label Property
12762	Fix	PopupMenuItem. Caption Property
12762	Fix	PopupMenuItem. HelpString Property
12762	Fix	PopupMenuItem. Macro Property
12762	Fix	PopupMenuItem. Type Property
16244	Fix	PlotConfiguration. PlotType Property

LISP

1 were added(highlighted by blue) and 23 were fixed as below.

Bug ID	Modification	Description
16233	Add	acet-ui-message
4513	Fix	open
8553	Fix	entmod
15668; 15881; 15882 ; 16830; 16391	Fix	command
15913	Fix	inters
15946; 16023	Fix	/
16197	Fix	prinl
16246	Fix	entmake
16541	Fix	getkword
16631; 17083	Fix	read-line
16677	Fix	trans
16801	Fix	write-line
16515	Fix	load
10614; 16679	Fix	load_dialog
16371	Fix	new_dialog
10710	Fix	unload_dialog
15980	Fix	vl-cmdf
16019	Fix	vl-string-position
16059	Fix	vla-get-mspace

16228	Fix	vl-bb-set\vl-bb-ref
16245	Fix	vlax-ename->vla-object
16257	Fix	vlax-dump-object
16288	Fix	vlax-safearray->list
16557	Fix	vlax-ldata-get

Bug Fixes

Bug ID	Description
API	
16557	lisp\vlax-ldata-get: Errors happen when executing operations on entities without extension data.
16382	The interface zrx\AcEdInputContextReactor:: beginQuiescentState won't be triggered when starting up ZWCAD and opening recent files.
9376	ssget\3dpoly: "Fit/Smooth" 3D polylines cannot be selected when executing "ssget" command in C mode.
Others	
16606	Help: Tutorial of wildcard is not available in Help document yet.
15736	Field: Fields will not updated after the original objects are deleted.

For the complete list of Bug Fixes, please refer to:

https://zwcad.freshdesk.com/a/solutions/articles/24000043141-what-s-fixed-in-zwcad-2020-official

Limitation and Notes

Bug ID	Description
17237	DATAEXTRACTION: When reselecting objects to extract data, the selection set are not reset.
17235	DATAEXTRACTION: The size of the dialog box cannot be changed.
17234	DATAEXTRACTION: Only one attribute can be extracted when a block contains 2 attributes of the same name.
17245	FLATTEN: The Z coordinate of the center of a circle will change when using “FLATTEN” command.