



infrADAC

Your solution to ADAC XML
made easy.

User Manual

Revision: 1.0



infrADAC
ACAD



infrADAC
BCAD

Developed By:



GALAXY PIXEL
STUDIOS

Change Log

Version	Date	Author	Comments
1.0.0	09/03/2026	Shai Whittaker	Initial Release.



Table of Contents

Change Log	1
Table of Contents	2
Glossary of Terms	4
Core concepts	4
Projects and attributes	4
Assets and libraries	4
Drawing, geometry and display	5
Files, supporting data and logs	5
Introduction:.....	6
Purpose of the Plugin	6
Key Features	6
System Feature Requirements	7
Where to download infrADAC	7
Installing infrADAC:	8
Before you start	8
Welcome screen	8
Licence Agreement	9
Licence Key	9
Choose CAD host location and profiles	10
Installing	11
Installation complete	12
Post-installation Configuration	12
Getting Started:.....	13
Initial Setup	13
Functionalities:	15
Creating a new project	17
Editing an existing project	18
Setting the Project Attributes	19
Setting the Global Attributes	20
Standard Assets	21



Supplementary Asset	26
Adding an Enhancement Asset	29
Handling Attribute Blocks	32
Cloning an Asset	32
Cleaning Assets.....	35
Drawing Extents.....	38
Producing an Audit Report.....	41
Audit Report Error Highlights	48
Exporting an infrADAC design to ADAC XML	49
Importing an ADAC XML file into model space	51
infrADAC Log	53
Updating a Licence	54
Windows and Forms:	56
Asset Library	56
Asset Profile Window	60
Settings.....	62
Software Updates Window	74
Update Logs Window	76
Select Licence File Window	80
Support Ticket Form.....	81
Asset XData Viewer.....	85
Find Asset Tool	89
Help Window	93
About Window	95



Glossary of Terms

Core concepts

- **infrADAC**: A CAD plugin designed to create and audit ADAC XML files so asset data matches the required ADAC XML schema.
- **ADAC**: “As Design and As Constructed”, a standardised specification for describing asset design and as-constructed data for transfer to asset owners.
- **ADAC XML**: The XML submission file created from a drawing during export.
- **ADAC XML schema**: The formal structure and validation rules that ADAC XML must follow.
- **CAD host**: The CAD application **infrADAC** runs inside, such as AutoCAD or BricsCAD.
- **Receiver**: The organisation that will receive the ADAC XML and defines required fields, standards and validation rules.
- **Profile**: A packaged configuration that controls the active standard, asset libraries and rule sets used for auditing and exporting.
- **Base Schema**: The selected base ADAC standard for a project that protects core definitions so they remain compliant.
- **Embedded mode**: A restricted state where no profile is loaded, so the library is view-only and adding or editing assets is blocked.

Projects and attributes

- **Project**: The drawing-level setup used by infrADAC for auditing and export, including required project metadata.
- **Project attributes**: Required project-level fields (metadata) that must be completed to meet the receiver’s requirements.
- **Hardcoded attributes**: Profile-enforced project fields that cannot be edited in the normal project attributes form.
- **Global attributes (prefill)**: Default values applied across assets so users stay consistent and reduce repeated data entry.
- **Required attribute**: A field marked as mandatory that must be set before audit and export can pass.
- **Enumeration values**: A controlled list of allowable values for an attribute used during validation.

Assets and libraries

- **Asset**: A drawing object that has ADAC data attached and will be included in audit and export outputs.
- **Asset definition**: The library template that describes an asset’s attributes, geometry requirements and export behaviour.
- **Asset library**: The list of available asset definitions provided by the active profile and any supplementary additions.
- **Standard asset**: A base asset defined by the selected ADAC standard and exported in the normal ADAC structure (not under Supplementary).
- **Supplementary asset**: An additional asset definition that extends the base library and exports under the Supplementary section with its own feature class and attributes.



- **Enhancement asset:** An informational asset (such as annotation or dimension outputs) that communicates design intent alongside the exported data.
- **Feature Class:** The class or group the receiver expects the asset to sit under (for example OpenSpace, Electrical, Transport).
- **Alias:** An optional short label used for display or naming conventions.
- **Library mode:** The library window state used to manage standard and supplementary asset definitions and related configuration.
- **Generate mode:** The library window state used to create assets from selected geometry and show what definitions are accepted or rejected.
- **Rejected asset:** An asset definition that is not compatible with the current selection or rules and is disabled for generation (often hidden unless Show Hidden is enabled).

Drawing, geometry and display

- **Model space:** The main drawing workspace where assets and geometry are created and edited.
- **Layer:** The CAD layer an entity is placed on which is often used for organisation and visibility control.
- **Geometry type:** The geometry form an asset expects (point, linear, area etc) which controls what it can be generated from and how it exports.
- **Drawing extents:** A boundary defined by SouthWest and NorthEast XY coordinates used to limit work areas and validate extents-related rules.
- **Prefill:** An automatically calculated value (for example length or area) derived from geometry and stored on the asset.
- **Drawing configuration:** The settings that control how an asset is drawn, including line appearance, optional blocks and attribute block behaviour.
- **Attribute block:** An optional block used to display asset attribute values in the drawing for quick viewing without opening other windows.
- **Dynamic block:** A block definition that supports multiple configurations through visibility states.
- **Visibility state:** A selectable state within a dynamic block that changes which elements and attributes are active or visible.
- **Attribute tags:** The names of block attributes used for validation and mapping to expected asset attribute names (often using a prefix convention).

Files, supporting data and logs

- **Supporting files:** External files associated with assets that can be referenced and packaged with outputs where applicable.
- **SupportingFiles cache:** A temporary local store used by **infrADAC** to manage supporting files for the current project and asset IDs.
- **ComponentInfo:** An exported metadata container that can include notes and supporting file



Introduction:

In the digital age, the accuracy and standardisation of asset data collection are paramount for effective infrastructure management. As urban areas expand and public utilities evolve, having precise and reliable data is essential for planning, maintaining and upgrading various infrastructure assets. The seamless transfer of asset information into council, government, and public/private utility systems ensures operational efficiency, compliance and informed decision-making.



Purpose of the Plugin

infrADAC is a specialised software tool designed to facilitate the design and auditing of ADAC XML files. ADAC (As Design and As Constructed) provides a standardised specification for describing asset design and as-constructed data, which is essential for the efficient transfer of asset information to councils, governments and utilities. The **infrADAC** plugin ensures that asset data conforms to the ADAC XML schema, supporting a range of infrastructure asset types, including transport, stormwater, open space, water, sewerage, communications, electrical, cadastre and surface.



Key Features

infrADAC offers robust functionality to streamline the management and auditing of ADAC XML files, ensuring accuracy and compliance with the ADAC schema. Key features include:

- **Profiles:** Allowing users to preset specific auditing settings tailored to the requirements of different asset owners, such as councils, governments, and public utilities.
- **Auditing Tool:** Assists asset owners in ensuring that submitted files are complete and contain all relevant data and required information.
- **Supplementary and Standard Assets Management:** Simplifies the creation and editing of supplementary and standard assets in a single, easy-to-navigate window, making it easier for asset owners to work with assets specific to their needs.
- **Direct Import of ADAC XML:** Enables users to import ADAC XML files directly into the software, converting the data into CAD geometry in real coordinate 2D/3D model space. This ensures seamless integration with existing data and allows for accurate visualisation and manipulation of asset information.
- **ADAC XML Design Creation and Export:** Allows users to create ADAC XML designs directly within their existing DWG CAD files, which can then be exported as ADAC XML, facilitating a smooth workflow from design to compliance.

With these features and functionalities, **infrADAC** streamlines the management and auditing of ADAC XML files, ensuring accuracy and compliance with ADAC standards.



System Feature Requirements

infrADAC is a lightweight plugin designed to seamlessly integrate with your existing CAD environment. The primary requirement for running **infrADAC** is the capability to run AutoCAD/BricsCAD. If your system can handle either, it will be more than capable of running the **infrADAC** plugin without any issues.

Minimum System Requirements

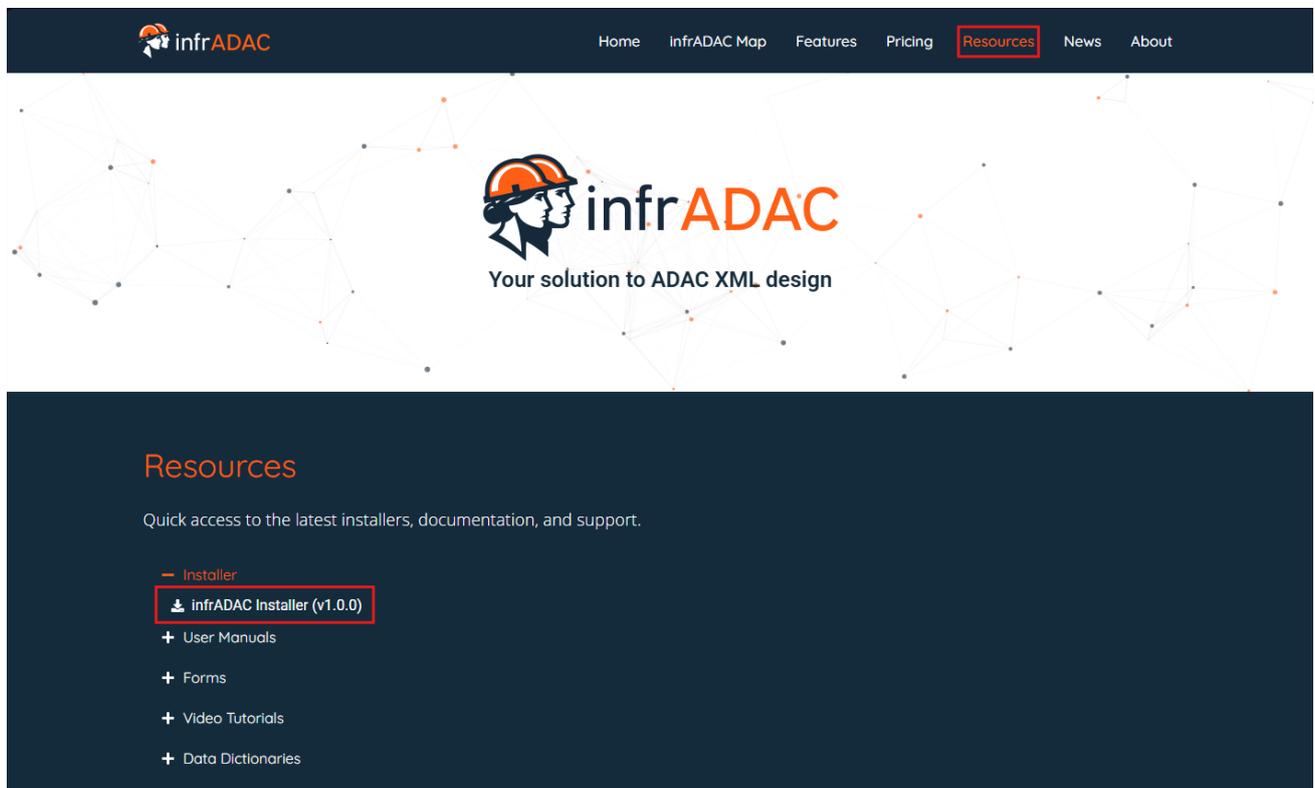
- **Operating System:** Windows 7 or later
- **Processor:** 2.5–2.9 GHz processor
- **Memory:** 4 GB RAM
- **Storage:** 50MB of free disk space
- **Graphics:** 1GB GPU
- **Software:** AutoCAD 2025 or later

Recommended System Requirements

- **Operating System:** Windows 10 or later
- **Processor:** 3+ GHz processor
- **Memory:** 6GB RAM or more
- **Storage:** 50MB of free disk space.
- **Graphics:** 4 GB GPU
- **Software:** AutoCAD 2025 or later

Where to download infrADAC

To download **infrADAC**, go to infrADAC.au/resources in your web browser. Scroll to the Installer dropdown, expand it, then download the latest available **infrADAC** Installer listed there.



The screenshot shows the website's navigation bar with links for Home, InfrADAC Map, Features, Pricing, Resources (highlighted with a red box), News, and About. The main content area features the infrADAC logo and the tagline "Your solution to ADAC XML design". Below this, the "Resources" section is displayed, providing quick access to installers, documentation, and support. A dropdown menu is open under "Resources", showing a list of items: "Installer" (with a minus sign), "infrADAC Installer (v1.0.0)" (with a download icon and highlighted by a red box), "User Manuals" (with a plus sign), "Forms" (with a plus sign), "Video Tutorials" (with a plus sign), and "Data Dictionaries" (with a plus sign).



Installing infrADAC:

The **infrADAC** installer wizard guides you through activation, host selection and installing the infrADAC application files, required profiles and optional packages and add-ons.

An internet connection is required because the installer activates your licence key and downloads the latest software release and any selected packages and add-ons.

Before you start

You will need:

- A valid **infrADAC** licence key
- An active internet connection
- A supported CAD host installed on the machine

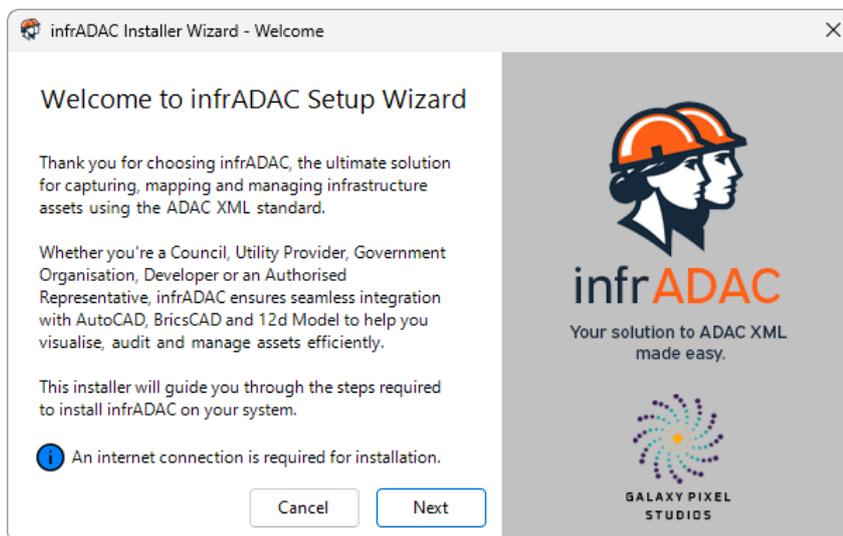
Host compatibility checks enforced by the installer:

- **AutoCAD** requires **AutoCAD 2025 or newer**
- **BricsCAD** requires **BricsCAD Pro 2026 or newer**
- **12d Model** requires **12d Model v15 or newer**

Welcome screen

Launch the installer. The Welcome screen explains what the wizard will do and reminds you that internet is required.

Click **Next** to continue or **Cancel** to exit.



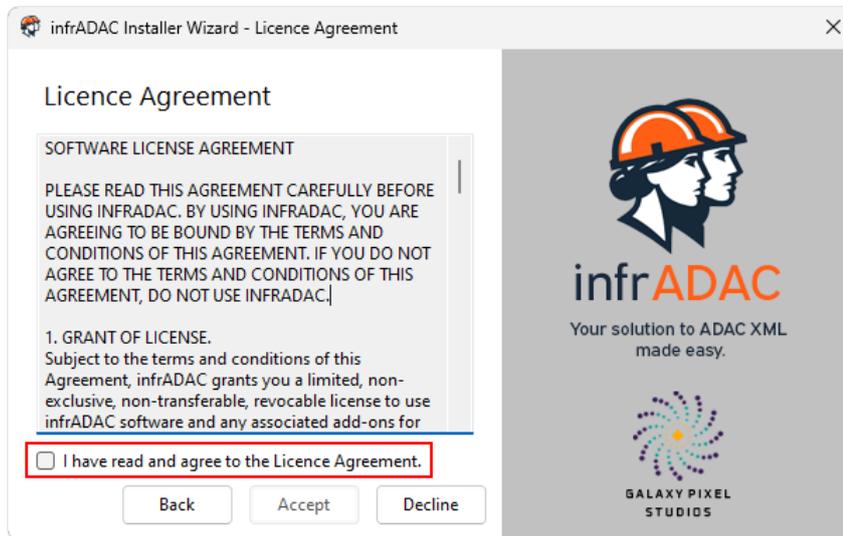
Licence Agreement

Read the licence agreement then tick:

I have read and agree to the Licence Agreement.

The **Accept** button is disabled until the checkbox is ticked.

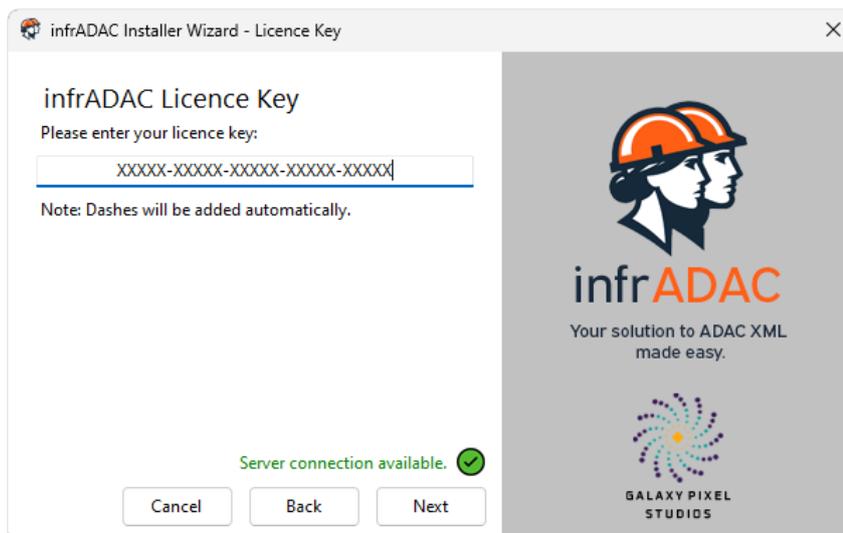
Click **Accept** to continue, **Back** to return to the Welcome screen or **Decline** to exit.



Licence Key

Enter your licence key into the textbox. The installer:

- Accepts alphanumeric characters
- Automatically formats the key with hyphens every 5 characters
- Enables **Next** only when the key is complete (XXXXX-XXXXX-XXXXX-XXXXX-XXXXX)



A connection indicator shows whether the installer can reach the activation server. If the server is unreachable you will see a red status  and you may need to check your internet connection before you can proceed.



What happens when you click Next?

When you click **Next**, the installer attempts to activate the licence key by sending:

- The licence key (normalised, without hyphens)
- Device identifiers (CPU, motherboard and Windows device ID)

If activation succeeds, the installer receives download information for:

- The **infrADAC** base software
- Any available packages
- Any available add-ons

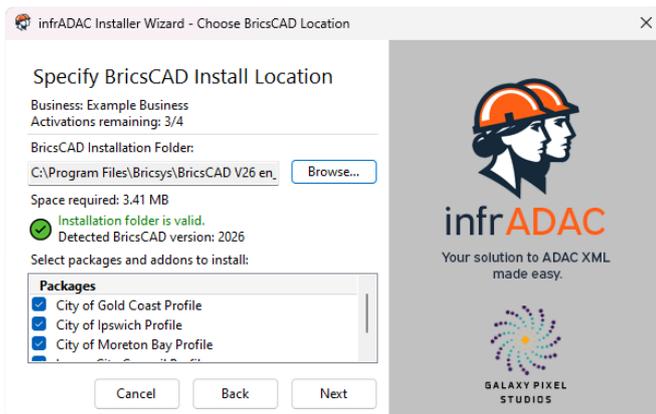
If activation fails, an error is shown with a support code you can quote to support.

Choose CAD host location and profiles

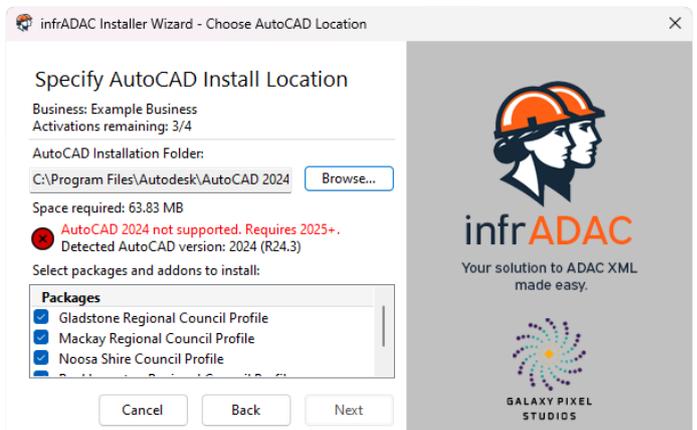
After activation, you will be asked to choose the installation folder for your CAD host (AutoCAD, BricsCAD or 12d Model).

1. Click **Browse** and select the CAD host install folder
2. The installer validates the folder by checking for the host executable:
 - AutoCAD: acad.exe
 - BricsCAD: bricscad.exe
 - 12d Model: 12d.exe
3. The installer detects the host version and shows it under the status message
4. Select which packages and add-ons to install (if provided by your licence)

BricsCAD: Compatible Example



AutoCAD: Incompatible Example



Where infrADAC is installed?

The installer creates an **infrADAC** root under the selected target base path:

<TargetBasePath>\GalaxyPixelStudios\infrADAC\

It then writes subfolders including Data, Licence, Logs, Lisp, Profiles, Projects, Uninstall and Update.

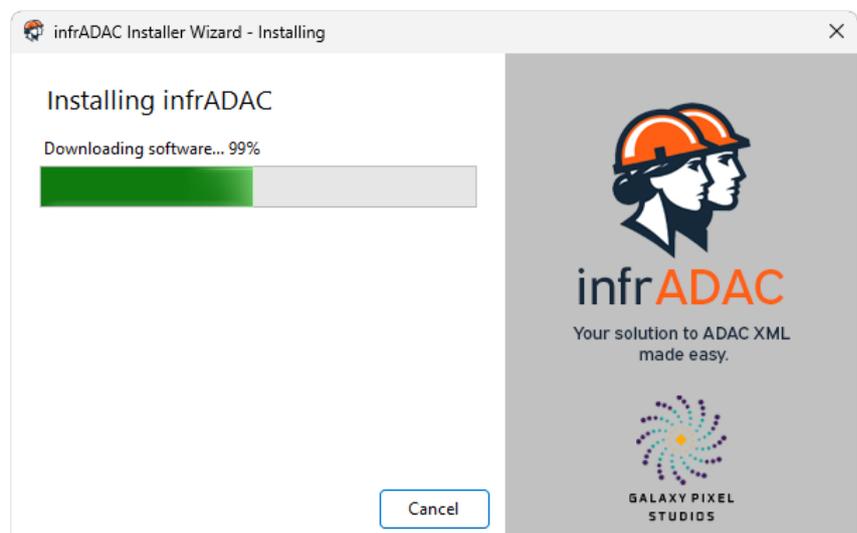
Important host specific behaviour:

- **AutoCAD:** the installer may resolve your selection into the per-user AutoCAD profile folder under LocalAppData and then writes the AutoLISP loader into the **Roaming** Support folder so it loads on document open
- **BricsCAD:** the installer installs into the per-user BricsCAD profile folder under LocalAppData (for example V26x64<locale>) and writes the AutoLISP loader into the **Roaming** Support folder using a Local to Roaming mapping

Installing

The Installing screen shows a progress bar and status updates. During installation the wizard:

- Creates **infrADAC** folders under GalaxyPixelStudios\infrADAC
- Writes Licence\licence.dat
- Authenticates with the download service
- Downloads the latest stable software release
- Verifies SHA256 (when provided)
- Extracts the software into the infrADAC folder
- Updates InfrADAC_Settings.json so Profile and Project folder paths match this installation
- Configures CAD startup auto-load for the infrADAC AutoLISP entrypoint (ADACRun.lsp)
- Registers the .idc file icon for the current user
- Downloads selected packages into the Profiles folder
- Downloads selected add-ons and either extracts them (zip) or copies them into the app folder
- Registers the uninstaller and updater for the current user

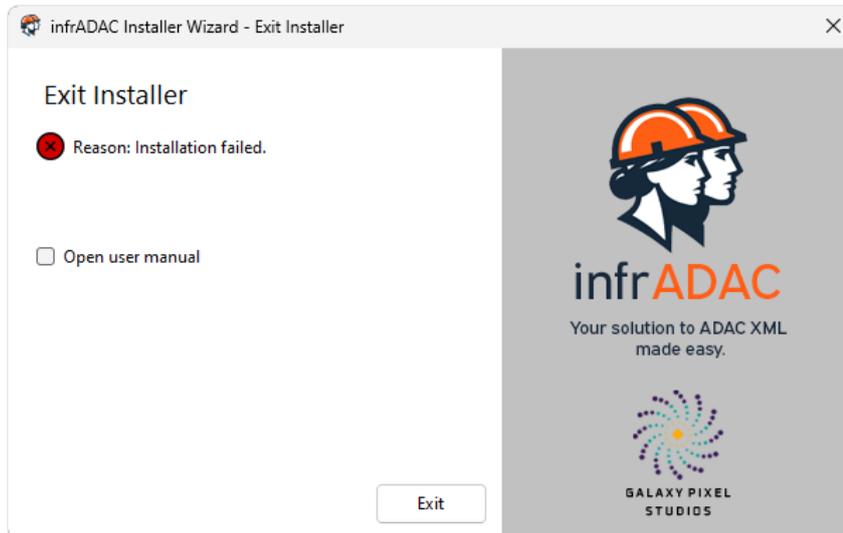


Installation complete

When complete, you will see the completion screen. Click **Finish** to exit. If the checkbox is ticked, the installer will open the user manual after you finish.

If the user cancels or an error occurs

If the installer is cancelled, the licence is declined or an error occurs, the Exit screen is shown with a reason message. The user can optionally open the manual before exiting.

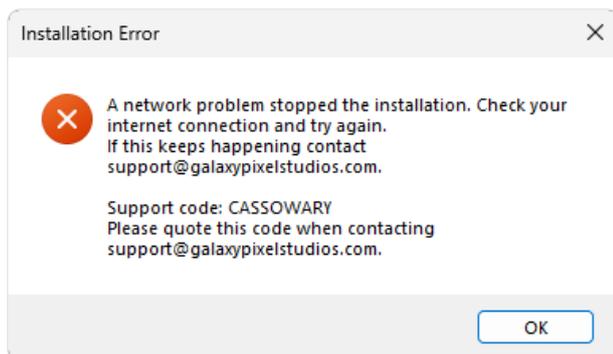


Troubleshooting and support codes

When the installer shows an error, it includes a deterministic support code (an Australian animal). Ask the user to quote that code to support so you can identify the error category quickly.

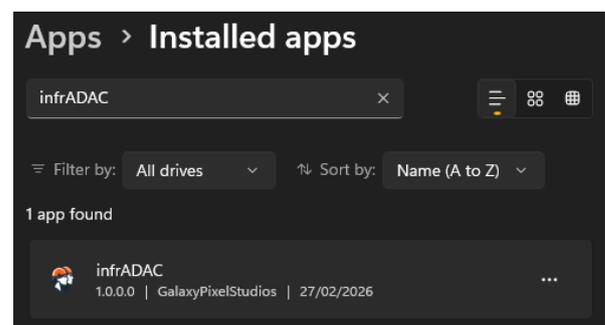
The installer also writes a log file here:

`%LOCALAPPDATA%\GalaxyPixelStudios\infrADAC\Installer\Logs\installer.log`



Post-installation Configuration

No post-installation configuration is required. The installer automatically deploys all required files and registers the add-in so it is ready to use on first launch. If you even need to uninstall infrADAC it can be done by running the uninstaller from **Installed apps** in Windows.

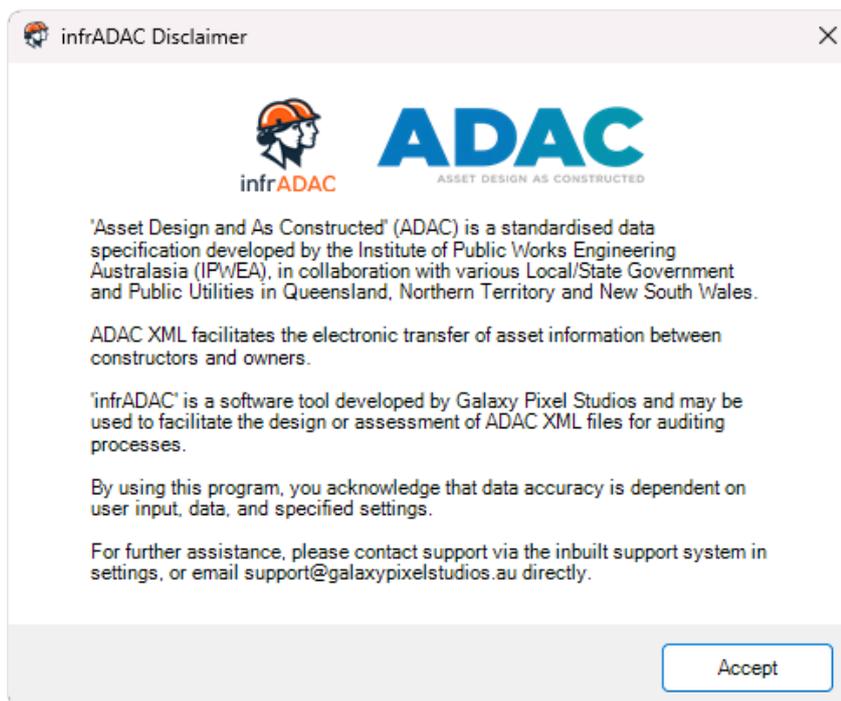


Getting Started:

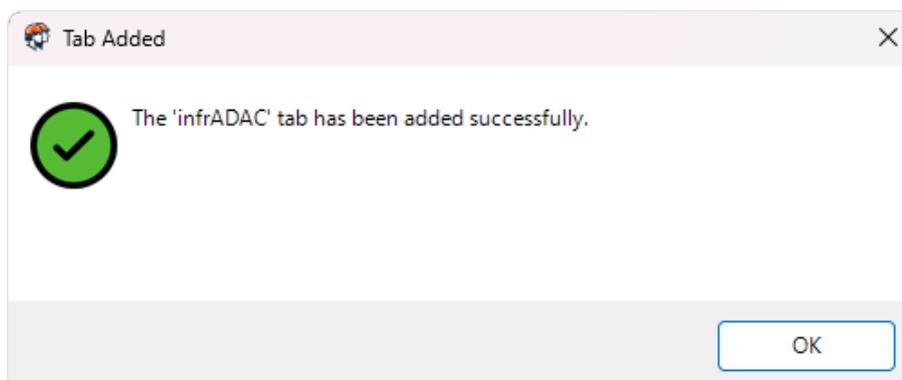
Initial Setup

During installation, a small LISP routine is installed to the startup suite that runs the NETLOAD command to start **infrADAC**. The installed version of AutoCAD or BricsCAD Pro will be identified so the LISP can be setup in the correct startup suite.

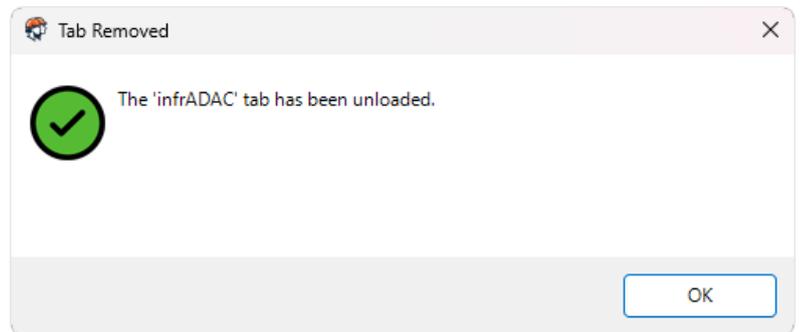
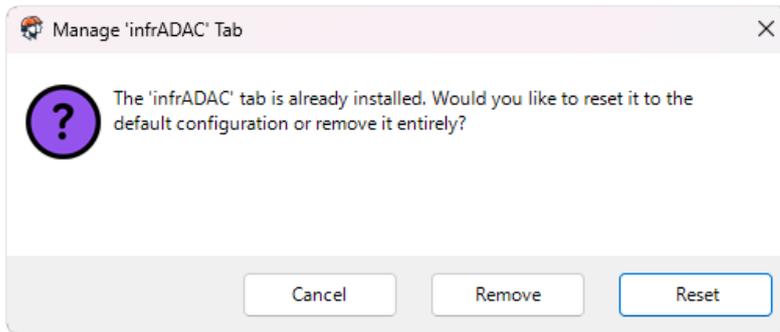
Once AutoCAD/BricsCAD Pro is open, type the command '**ADACRun**'  to begin the plugin. Users will be greeted with a splash screen and disclosure statement to be accepted. The plugin will not proceed until the disclosure statement is accepted. The statement will reappear when the command '**ADACAbout**'  is run.



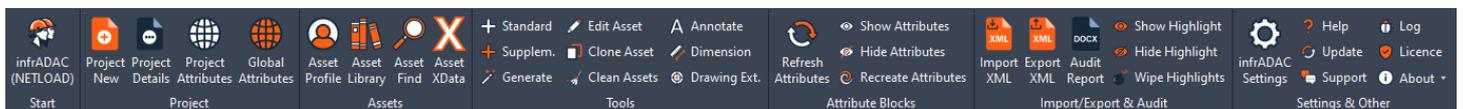
After accepting the disclosure statement, run the command '**ADACCUIX**'  to create the ribbon bar tab for **infrADAC**. The message box below will appear when the ribbon bar tab has been successfully added to AutoCAD.



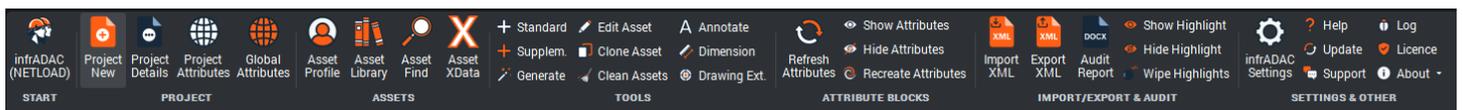
If there are any issues with the ribbon bar tab and the user wishes to reset or uninstall it simply run the 'ADACUIX'  command again and select the action to proceed with.



AutoCAD: 



BricsCAD: 



Functionalities:

infrADAC Commands

Command	Icon	Description
ADACRun		Starts the infrADAC Plugin. Requires the AutoLISP to be running.
ADACCUIx		Installs the infrADAC ribbon tab in AutoCAD or lets you reset it (reload) or remove it.
ADACAbout		Opens the Disclaimer / About window.
ADACHelp		Opens the infrADAC Help window (modeless so it can stay open while you work).
ADACSettings		Opens the infrADAC Settings window.
ADACLicence		Checks your licence status. If valid it shows licence holder and expiry, if not it prompts you to load a valid licence file.
ADACUpdate		Opens the Update / Check for Updates window.
ADACLog		Opens the latest log file to help diagnose issues.
ADACSupport		Opens the Support Ticket window to submit a support request.
ADACNew		Initialises a new infrADAC project in the drawing (layer, units and project/global data setup). This is the main “start here” command.
ADACProject		Opens the Project window to view or update project name, description and export file path.
ADACProjectAttributes		Opens Set Project Attributes to update project-level required fields. Requires ADACNew first.
ADACGlobalAttributes		Opens Set Global Asset Attributes to update global defaults used across assets. Requires ADACNew first.
ADACDrawingExtents		Sets and manages the project Drawing Extents area (creates or reconciles the extents rectangle and saved extents data).
ADACLibrary		Opens the ADAC Asset Library to review assets and adjust attribute and audit requirements.
ADACFind		Opens the Find Asset so you can search for an asset by its type and ADACId/ObjectId to jump to it in the drawing.
ADACProfile		Opens the Profile Editor for the currently loaded infrADAC profile.
ADACStandard		Adds a Standard ADAC asset to the drawing by choosing from the standard library then placing it.

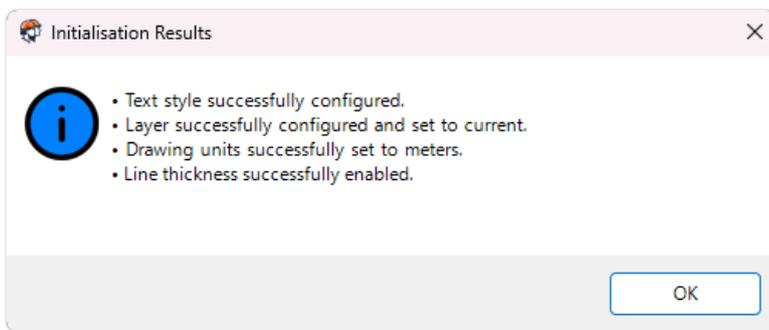


ADACSupplementary		Adds a Supplementary asset to the drawing by choosing from the supplementary library then placing it.
ADACGenerate		Generates an infrADAC asset from existing geometry: select objects, choose a compatible asset type, generate the asset.
ADACAnnotate		Creates an Annotation enhancement from the Enhancements library.
ADACDimension		Creates a Dimension enhancement from the Enhancements library (can use a single preselected object when applicable).
ADACEdit		Edits the attributes (data fields) of selected infrADAC asset(s).
ADACClone		Creates a copy of the selected asset including its infrADAC data.
ADACShowAttributes		Shows asset properties blocks (all if nothing selected, otherwise only for the selection).
ADACHideAttributes		Hides asset properties blocks (all if nothing selected, otherwise only for the selection).
ADACRefreshAttributes		Refreshes asset properties blocks so displayed values match current data (all or selection).
ADACRecreateAttributes		Recreates missing asset properties blocks for selected assets (handles grouped assets).
ADACClean		Repairs asset data by cleaning/renumbering ADAC IDs and rebuilding missing properties blocks (extents or whole model space).
ADACErrorShow		Shows error highlighting overlays (selection only if selected, otherwise all).
ADACErrorHide		Hides error highlighting overlays (selection only if selected, otherwise all).
ADACErrorWipe		Clears error highlighting overlays (selection only if selected, otherwise all).
ADACImport		Opens the Import window to bring infrADAC data into the current drawing.
ADACExport		Exports infrADAC data from the drawing after checking required project attributes. Requires ADACNew first.
ADACAudit		Opens the Audit window to review infrADAC content for issues and compliance.
ADACXData		The ADACXData command opens the infrADAC Asset XData Viewer window so you can inspect the XData attached to model space entities.
ADACReference		Open the project reference file for the latest base schema version. Displays all available assets per class.
ADACExample		Opens an example project file to provide a reference of a compliant ADAC XML design created with infrADAC.

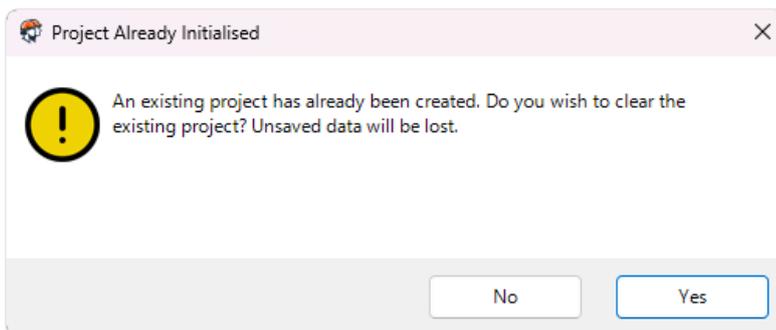


Creating a new project

To create a new project run the **'ADACNew'**  command. This will run an initialisation method that performs a number of setup changes to the drawing files model space. None of these changes are necessary for the plugin to work; they serve only to improve the functionality of the ADAC XML design process. To remove any of these changes from the initial project setup you can go to the **Settings Window** → ADAC New and select/deselect the actions. By default, all actions are enabled. A window will appear with the initialisation results

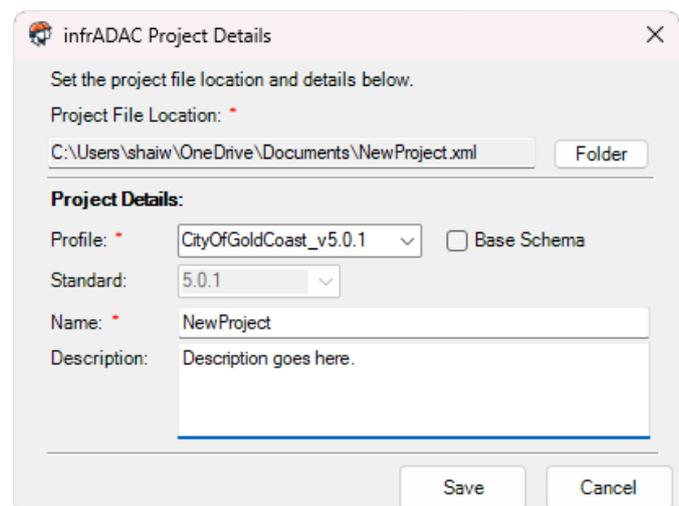


When running the **'ADACNew'**  command after a project is already created, the user will be prompted to clear the existing project information before proceeding with the new project details.



In both cases, after proceeding with the **'ADACNew'**  command, the project details window will appear for users to define some basic information about the project. The user can define the expected xml file location (which can be reviewed and adjusted on export), the desired profile/ standard they wish to design the ADAC XML file to as well as the name and description of the project.

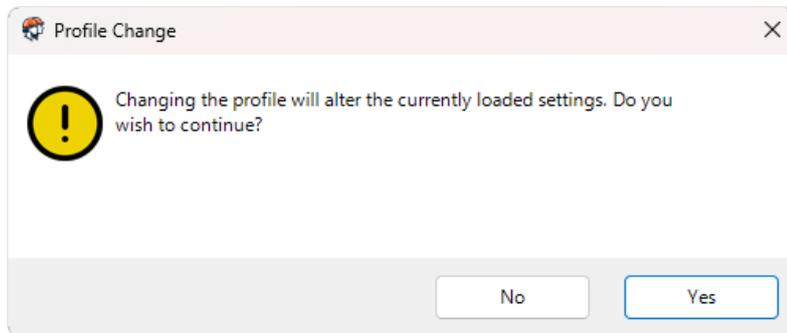
If the user wishes to utilise a built-in base ADAC XML standard they can select the 'Base Schema' checkbox which will enable the 'Standard' drop down for the user to select the desired version of they wish to design to. Selecting this option will not allow the user to change any attribute



Editing an existing project

The 'ADACProject'  command is used to review and update the existing project setup for the current drawing. When opened, the form automatically prefills the current project name and description from the stored project attributes and reloads the current export file path if one has already been set. This allows users to edit the existing project details rather than creating them again from scratch. The form also reflects the currently loaded profile and its associated ADAC standard.

Users can update the project file location, project name, description and profile as needed. If the profile is changed, the software prompts the user to confirm the change before loading the newly selected profile and updating the active standard. By default, the displayed standard follows the selected profile, but if **Base Schema** is enabled the standard can be manually adjusted. When the user saves, the updated project name, description and export file path are written back to the drawing and settings. If the project name has changed, the associated temporary SupportingFiles project folder is also renamed to match. If the form is cancelled, no changes are saved.



Setting the Project Attributes

Depending on the profile currently loaded, some project level attributes are considered hardcoded and cannot be adjusted directly in this form. This ensures compliance with the receiver’s export requirements and should reflect the standard issued by the receiver. Where changes are needed, the Project Attributes window provides the controlled location for reviewing and updating project level information for the current job.

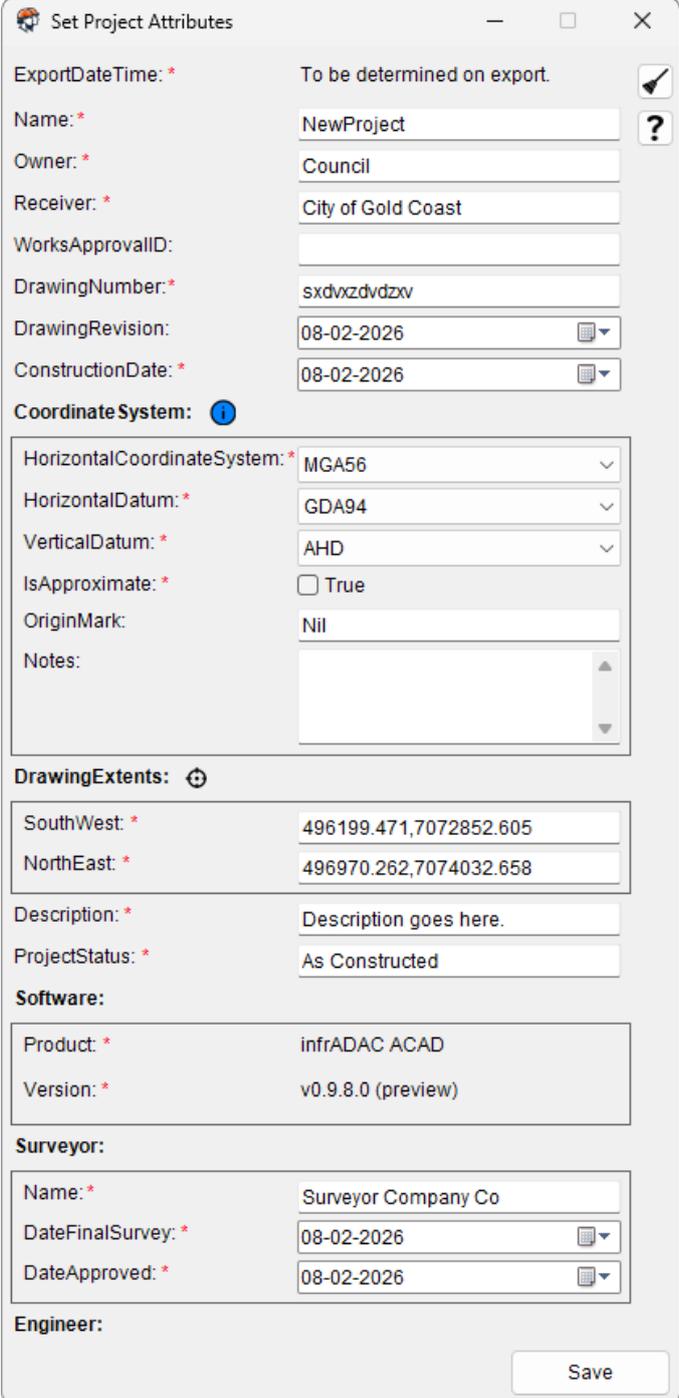
Any attributes that display a red asterisk symbol * are mandatory. Users will not be able to successfully complete export or auditing processes until those required fields have been populated. The form also applies a maximum display height, so some properties may not be visible immediately when the window opens. In these cases, users can scroll through the form to access the remaining fields.

The Project Attributes window is only available after the drawing or job has been initialised using the ‘ADACNew’  command. If the system has not yet been initialised, the command will not open and the user will be prompted to complete setup first. The command also checks licence and disclaimer status before allowing access.

When opened, the form reads the Project attribute definition from the currently loaded profile. If a user-specific default asset definition exists for the active profile, that definition is used in preference to the base default asset definition. This means the available project fields can vary depending on the loaded profile and any configured user overrides.

When the form is reopened, previously saved project attribute values are prefilled back into the window where available, allowing users to review and update existing project information rather than re-entering all values from scratch. If the user confirms the form, the project attributes are updated for the active drawing. If the user cancels the form, no changes are applied.

Users can reopen the form at any time by typing the ‘ADACProjectAttributes’  command to review or update project details as needed.




Setting the Global Attributes

Being able to predefine the global attributes assigned to each asset before it is created allows users to focus on the asset-specific information while maintaining consistency across the design. Global attributes define the standard information structure for the asset and help ensure required data is captured in a consistent and controlled way.

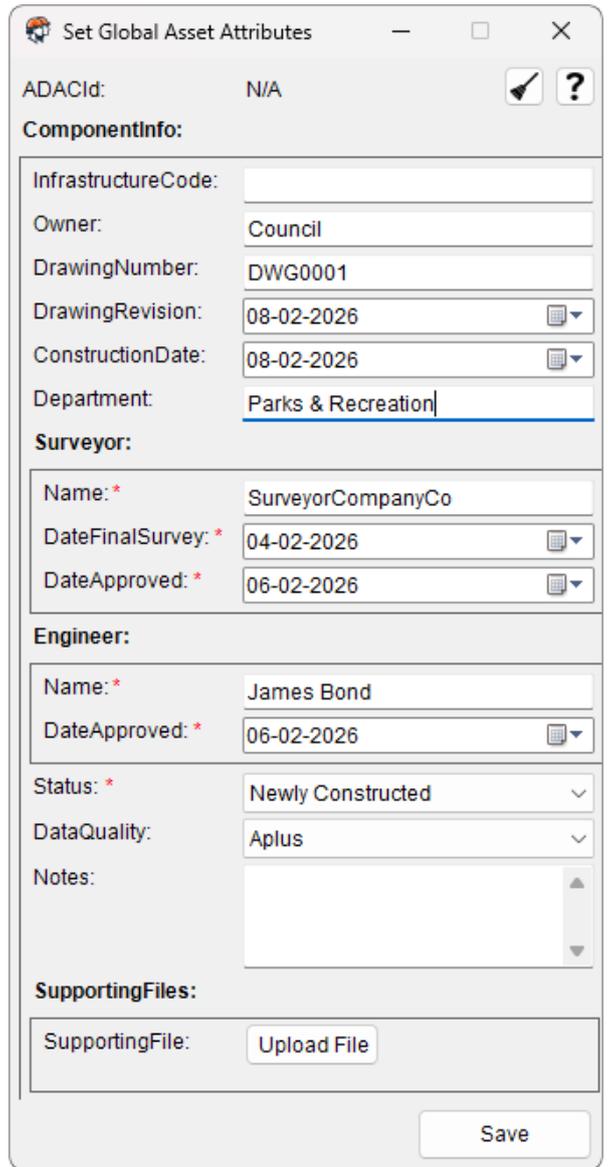
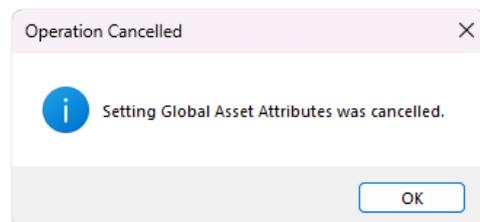
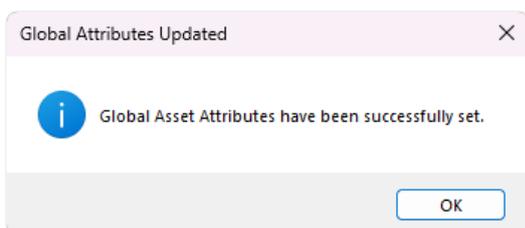
Within the Global Attributes window, users can search for existing attributes, review their details, and add, edit, clone, delete or reorder attributes as required. Each attribute can be marked as Required and assigned an appropriate data type, including numeric, text, boolean, date, time and enumeration. Text attributes can also be configured as multiline where needed, and specific text types can enforce maximum character limits or require a non-empty value.

Where appropriate, attributes can be further controlled by predefined allowable values. Depending on the selected type, this may include a fixed value, a valid value range or a selectable list of options. For enumeration values nested attributes can also be defined to capture additional information when particular values are selected. These nested attributes are displayed in separate tabs to make them easier to review and manage.

Validation is applied to help maintain data quality. Values must match the selected attribute type, and ranged entries must use a valid start and end value. Boolean fields can be configured to display as either True/False or Yes/No. Where applicable, numeric attributes such as length or area can also be configured to prefill automatically from the asset geometry.

Users should be aware that changing an attribute type may clear any existing fixed values, ranged values or selectable options associated with that attribute. When this occurs, the system prompts the user to confirm the change before the existing setup is removed. A preview function is also available so the attribute layout can be reviewed before saving.

Changes to global attributes are only applied once they are saved. Closing or cancelling the window before saving will discard any unsaved changes.

Standard Assets +

Standard assets are the base assets defined by the given ADAC XML schema loaded by the profile.

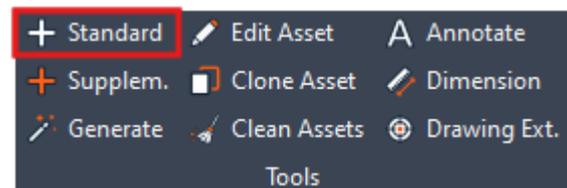
Geometry data is not generated until export, so even after defining an asset and adding it to the model space users are free to continue adjusting the assets its geometry and position. It should be noted that some assets that include attributes like area or length may need to be manually adjusted by editing the drawing if manipulated after being created.

Safeguards are ultimately ensured by the **Audit Report** that is performed before exporting the ADAC XML file is allowed. Unintended adjustments to assets that may cause inconsistencies with geometry or details at export should be caught before the file is created for submission.

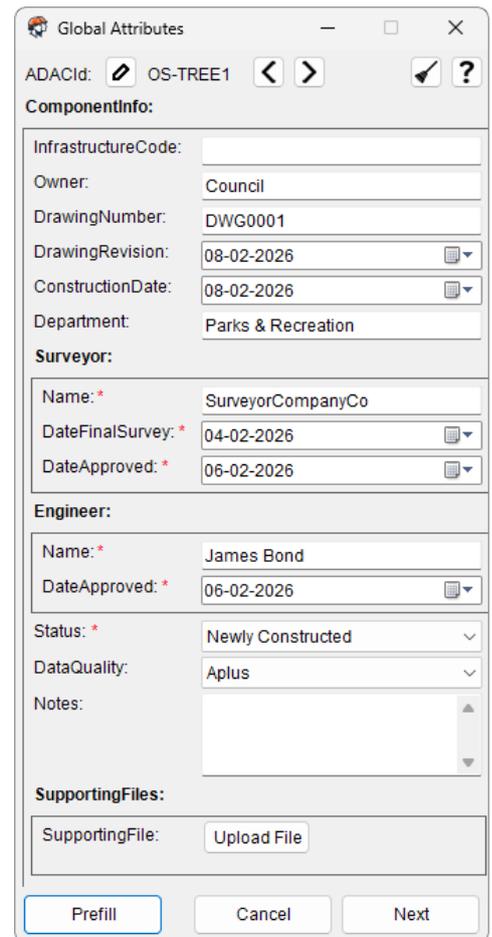
Standard assets are the base assets defined by the ADAC XML schema loaded by your profile. They export into the normal ADAC category and feature class structure for that standard (not under Supplementary).

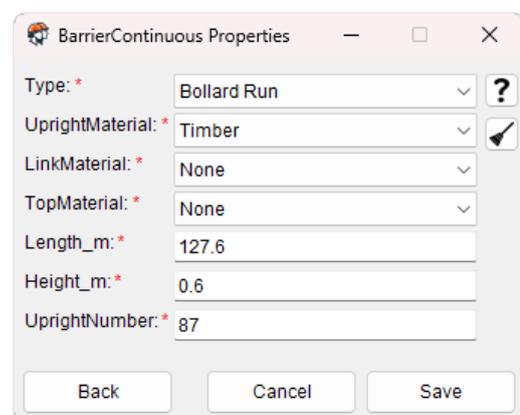
Adding a Standard Asset to the design

1. Open the Asset Library (Library Editor) with the 'ADACLibrary'  command.
2. The Asset Library Window will open with the list of available Standard Assets to add to the model space. The user can filter results by name, feature class or by typing key words into the search bar for an adjusted list result.
3. When the asset to be added has been selected from the list click the Create button.
4. The system determines the geometry type defined for the asset and automatically starts the appropriate creation method.
 - Point assets are created as block objects.
 - Linear/Segment assets are created as line objects.
 - Path or area assets are created as polyline objects (open or closed).
5. The asset layer defined in the asset configuration is resolved. If the layer does not already exist in the drawing it is created automatically before the asset is inserted.
6. Create or nominate the asset geometry.
 - For point assets, select the insertion point in the drawing. If the asset supports multiple placements you can continue selecting additional insertion points until finished.
 - For line assets, either select an existing line or two-vertex polyline or draw the line by selecting a start point and end point.
 - For path or area assets, either select an existing line or polyline or draw the required polyline geometry. Where the asset configuration does not allow curves, curved polylines cannot be used.
7. Once the geometry is created the system writes the core asset information to the object. This includes the asset ID, category, feature class, geometry type and grouping information if multiple assets were created.
8. If multiple assets are created in a single operation they are automatically grouped together and assigned a group identifier.



9. The Global Attributes form is displayed. Enter the required global attribute values and confirm to continue.
10. After the global attributes are accepted the Specific Attributes form for the selected asset type is displayed. Enter the required asset-specific values and confirm to continue.
11. Some fields may be automatically prefilled based on the geometry or previously entered values where this behaviour is defined in the asset configuration.
12. If either attribute form is cancelled the asset creation process is cancelled and any geometry created during the command is removed.
13. After the attribute forms are completed successfully the system inserts the properties block associated with the asset. This block is linked to the asset and is used to display the asset attribute information in the drawing.
14. The properties block is placed near the first asset object and is assigned to the same layer as the asset.
15. The visibility state of the properties block is updated to match the asset type configuration and the asset creation process is completed.





Asset Creation Behaviour

Standard assets are created using the geometry and configuration assigned to the selected asset type, so the user does not manually choose whether the asset is inserted as a block, line or polyline. That behaviour is determined automatically by the asset definition. Some assets allow only a single placement while others allow multiple placements within the same command. For assets that support multiple placement, the user can continue placing geometry until the operation is finished.

The asset is not fully created until the required Global Attributes and Specific Attributes forms have both been completed and accepted. If the process is cancelled before those steps are confirmed, the asset will not be finalised. If no valid geometry is created or selected, no asset is added to the drawing. Where user-specific block or line type overrides are available, those will be used instead of the default profile files.



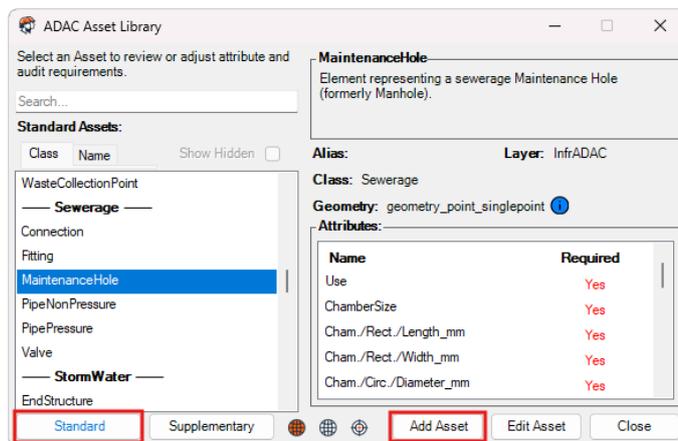
Creating a new Standard asset

Before you start

You must have a profile loaded to add, edit or delete assets. If you are in embedded mode (no profile loaded) the library is view only and the program will show a warning that functionality is limited.

Step by step

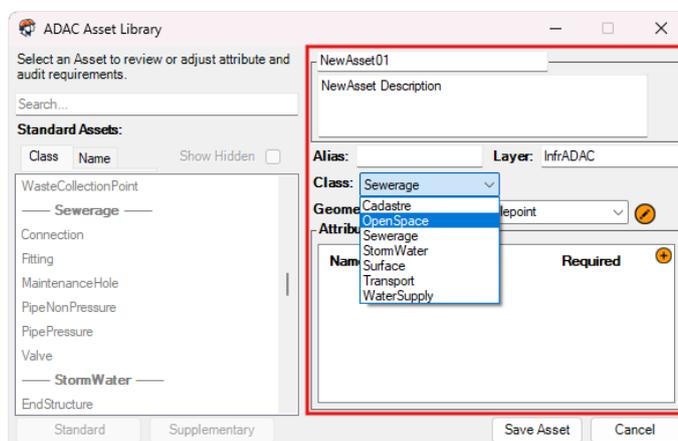
1. Open the Asset Library with the 'ADACLibrary'  command.
2. Click Standard at the bottom of the window to switch the list to Standard assets.
3. Click Add Asset.
4. The form switches into edit mode and shows default values for a new asset.



Enter the details

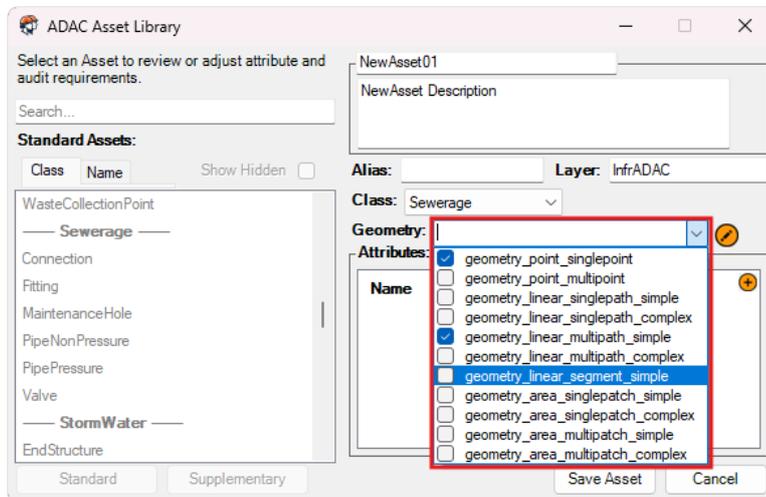
You will typically fill in the following fields:

- Asset Name: A clear, unique name for the asset (for example Tree, Pit, Pole, Switchboard).
- Description: A short description that helps users identify the asset.
- Category: The Standard category the asset belongs to (for example WaterSupply, Sewerage, StormWater, Transport, OpenSpace, Electrical depending on the loaded profile).
- Alias (optional): A short label used for display or naming conventions.
- Layer (optional): The drawing layer this asset is intended to use. If you leave it as default, it will generally remain **infrADAC**.



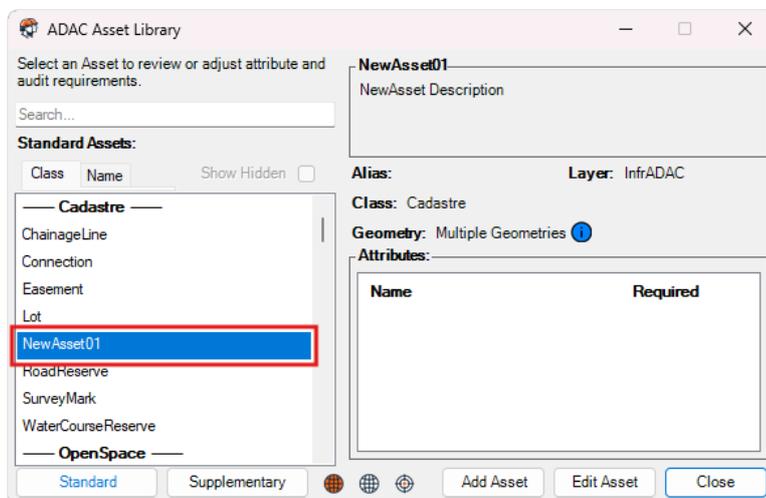
Choose the Geometry

Open the Geometry selector and choose the geometry type that matches how the asset will be drawn in the design. If the geometry selector supports multi selection, you can tick more than one geometry type. When more than one is selected, the field will show Multiple Geometries.



Save the asset

1. Click Save Asset.
2. If the save is successful, you will return to view mode and the new asset will appear in the list under Standard assets.



What happens after you click Save?

After saving a new asset, the program may ask two setup questions:

1. **Would you like to add attributes to the new asset?**

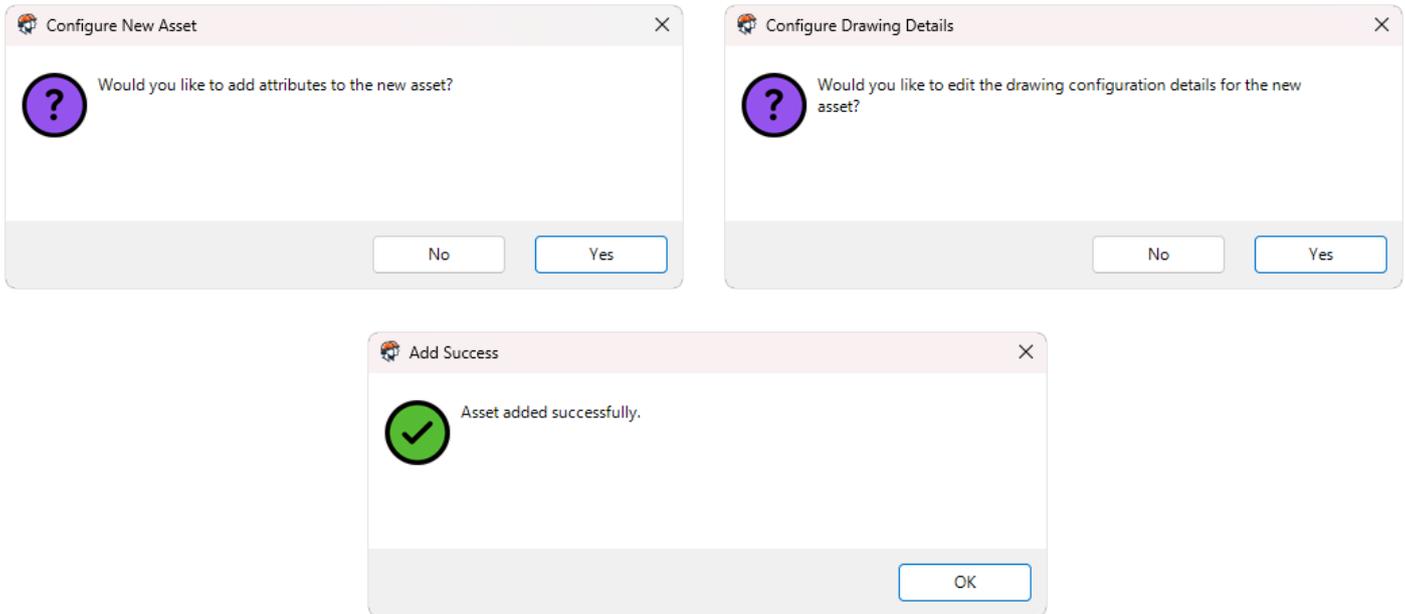
Choosing Yes opens the attribute editor so you can define the fields users will fill out on placement or later edits.

2. **Would you like to edit the drawing configuration details for the new asset?**

Choosing Yes opens the drawing configuration editor where you can set how the asset is represented in the drawing (for example block style for point assets or line style for linear assets).



You can choose No to both and come back later using Edit Asset.



Editing and deleting Standard assets

Standard assets fall into three practical groups in the library:

Base Standard assets (from the loaded ADAC profile)

These are the official schema assets. You can view them and, in some cases, create a user override for project needs but they cannot be permanently removed from the base library.

Standard assets (User override)

If you edit a base Standard asset, the program stores your changes as a user override. In the library, this can show as a user file entry and the Delete button may appear as Reset so you can revert back to the base Standard definition.

Additional Standard assets

These are extra Standard assets you add through Add Asset. They behave like Standard assets in export but they are not part of the original embedded standard, so they can be deleted.

Validation rules

Save Asset will be blocked if any of these are missing:

- Asset Name is blank.
- Category is blank.
- Geometry is blank.

You also cannot create a duplicate using the same combination of Asset Name and Category. If that combination already exists, you must change either the name or the category.



Tips and common issues

I cannot add assets

A profile must be loaded. If you are in embedded mode, the Add Asset button is blocked and the library is view only.

Duplicate asset message

You already have an asset with the same Asset Name and Category. Rename the asset or choose a different Category.

My geometry selection does not match how I want to draw it

Edit the asset and adjust the Geometry. If you are unsure, keep it simple and pick the one geometry type you actually intend to draw in the design (point for blocks, linear for lines, area for polygons).

Supplementary Asset +

Supplementary assets let you extend beyond the base ADAC asset library so you can include additional asset types required by your project. They behave like standard assets in the drawing but are exported under the Supplementary section of the ADAC XML with their own Feature Class and attributes.

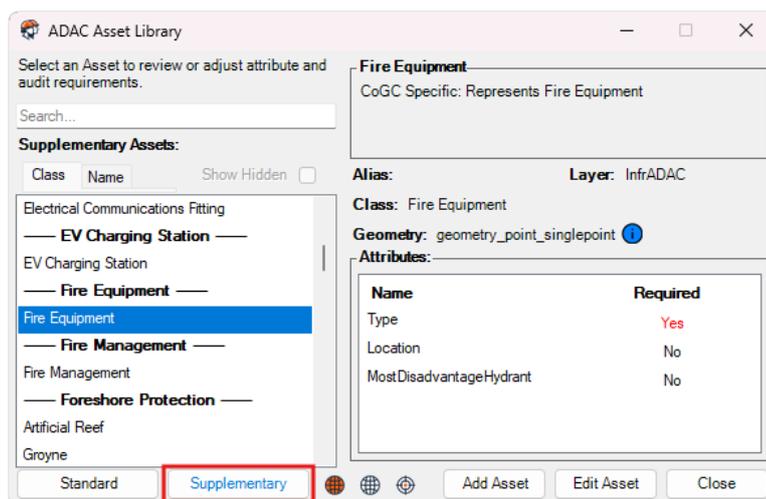
Before you start

A profile must be loaded to add, edit, delete or clone assets. If you are in embedded mode with no profile loaded the library is view only and **infrADAC** displays a warning that functionality is limited.

Where to add a Supplementary Asset

Supplementary assets are created in the Asset Library editor.

1. Open the Asset Library (Library Editor).
2. Select Supplementary at the bottom of the window to switch the list to Supplementary assets.



Create a new Supplementary asset

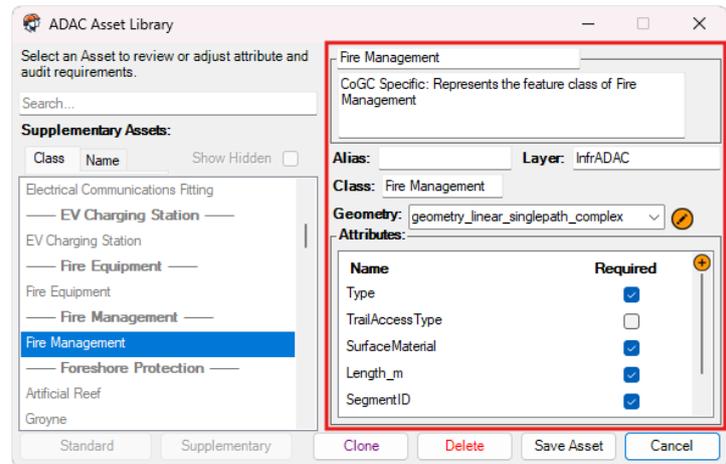
Click Add Asset. The form switches into edit mode and displays placeholder values for a new asset.



Enter the basic details

You will typically fill in:

- **Asset Name:** A clear, unique name for the asset (for example Tree, Pit, LightingColumn).
- **Description:** A short description that helps users identify the asset.
- **Feature Class:** The class or group the receiver expects this asset to sit under (for example OpenSpace, Electrical, Transport depending on your project conventions).
- **Alias (optional):** A short label used for display or naming conventions.
- **Layer (optional):** The drawing layer the asset is intended to use. If left blank it will typically default to **infrADAC**. When set it is also applied to the drawing configuration for the asset.



Choose the geometry

Open the Geometry selector and choose the geometry type that matches how the asset will be drawn. The available options follow the ADAC geometry tokens (for example `geometry_point_singlepoint`, `geometry_linear_singlepath_simple`).

The Geometry selector supports multi selection. When more than one geometry is selected the field displays Multiple Geometries and the tooltip shows the full list.

Save the asset

If the save is successful the form returns to view mode and the new asset appears in the list.

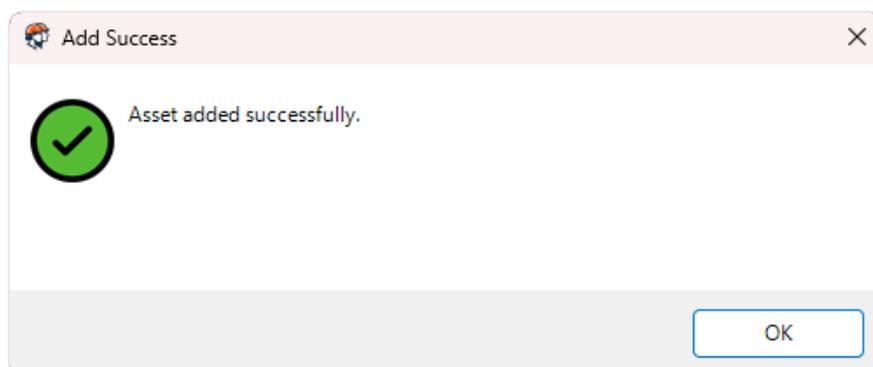
After Saving a New Asset

After saving a new asset the program may ask two setup questions:

Would you like to add attributes to the new asset? Choosing Yes opens the attribute editor so you can define the fields users will fill out when placing or editing the asset.

Would you like to edit the drawing configuration details for the new asset? Choosing Yes opens the drawing configuration editor where you define how the asset is represented in the drawing (for example block style for point assets or line style for linear assets).

You can choose No to both and come back later using Edit Asset then selecting the Attributes editor or Drawing Configuration editor for the asset.



How Supplementary assets appear in the exported ADAC XML

When exported, supplementary assets are written under the Supplementary section of the ADAC XML.

Each supplementary asset is exported with:

- ADACId (the unique ID for the placed asset).
- ComponentInfo (global metadata such as SupportingFiles and Notes, if present).
- Class (this is the Feature Class).
- Note (optional free text note).
- Attributes (your custom fields for the asset).
- Geometry (based on what was drawn).

What the Attributes tag means

In the Asset Library each Supplementary asset has an Attributes definition. This is where you set up the custom fields that users can fill out (for example Manufacturer, Model, SerialNumber, Rating).

When you export, those custom fields are grouped under a single ADAC:Attributes element for that supplementary asset. If your custom fields include nested paths or grouped fields the export nests them to match the structure you defined in the library.

Example XML structure for a Supplementary asset

This is a simplified example of how a point based supplementary asset is structured when exported. Your attribute names will match what you defined in the asset's Attributes section.

```
<ADAC:Supplementary>
  <ADAC:PointFeature xsi:type="StreetLight">
    <ADAC:ADACId>EL12</ADAC:ADACId>
    <ADAC:ComponentInfo>
      ...
    </ADAC:ComponentInfo>
    <ADAC:Class>Electrical</ADAC:Class>
    <ADAC:Note>Any free text note stored on the asset</ADAC:Note>
    <ADAC:Attributes>
      <ADAC:Manufacturer>ExampleCo</ADAC:Manufacturer>
      <ADAC:Model>XYZ-100</ADAC:Model>
    </ADAC:Attributes>
    <ADAC:Geometry>
      ...
    </ADAC:Geometry>
  </ADAC:PointFeature>
</ADAC:Supplementary>
```

Validation rules you need to know

Save Asset is blocked if any of these are missing:

- Asset Name is blank.
- Feature Class is blank.
- Geometry is blank.

You also cannot save a duplicate using the same combination of Asset Name and Feature Class. The check applies across both visible and hidden entries so you may need to change either the name or the class.

Tips and common issues

I cannot add assets

A profile must be loaded. In embedded mode the Add Asset button is disabled and the library is view only.



Duplicate asset message?

An asset with the same Asset Name and Feature Class already exists. Rename the asset or choose a different Feature Class.

My geometry selection does not match how I want to draw it.

Edit the asset and adjust Geometry. If you are unsure, choose the one geometry type you intend to draw (point for blocks, linear for lines, area for polygons).

Faster setup

Use 'ADACClone'  on an existing supplementary asset to copy its attributes and drawing configuration, then rename it and adjust only what is different.

Adding an Enhancement Asset

These asset types mainly serve as information and are predicated on the capability of the GIS or specific mapping system utilised to view the design. Enhancement assets are defined as either an annotation or dimension. **infrADAC** utilises the capability of the CAD system to

A specific case where enhancement assets may be required is providing offset dimensions for a water service or pole position relative to closest adjoining boundary.

Unlike typical **Standard** and **Supplementary** type assets, users cannot edit an enhancement asset after creating it and will need to delete and recreate the object in model space to assign the ADAC XML attributes it requires.

Purpose of Enhancement Assets

Enhancement assets are informational objects that help communicate design intent when viewed in a GIS or other mapping system. They are defined as either an annotation or a dimension. **infrADAC** uses the CAD system to create these objects and then attaches the required ADAC XML attributes to them.

A common use case is providing offset dimensions for a water service or pole position relative to the closest adjoining boundary.

Important: Unlike standard and supplementary assets, enhancement assets cannot be edited after creation.

To change an enhancement asset, delete the object in model space and recreate it so the required ADAC XML attributes can be assigned again.

When to use enhancement assets

- Provide offset dimensions and measurement context that must be preserved in export
- Add design notes, warnings, or callouts that need to travel with the exported dataset
- Communicate information that is not a physical asset but is still useful to downstream systems

Before you start

- You must have a valid licence and have accepted the disclaimer prompts (commands will exit if validation fails)
- The **infrADAC** layer must exist or **infrADAC** must be able to create it
- Enhancement definitions must be available in the active JSON configuration so the commands can load the asset schema

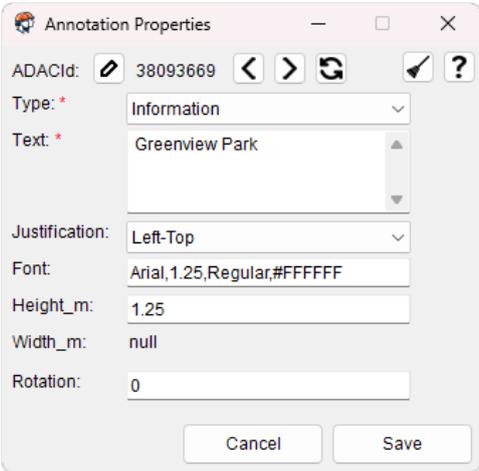


Create an Annotation Asset

Use 'ADACAnnotate'  to create an annotation (MText) enhancement and attach the required attributes.

- Run the 'ADACAnnotate'  command
- When prompted, pick an insertion point in model space
- Edit the text in the in-place MText editor then close the editor to continue
- Complete the attribute form and click OK to store the enhancement asset data on the annotation object

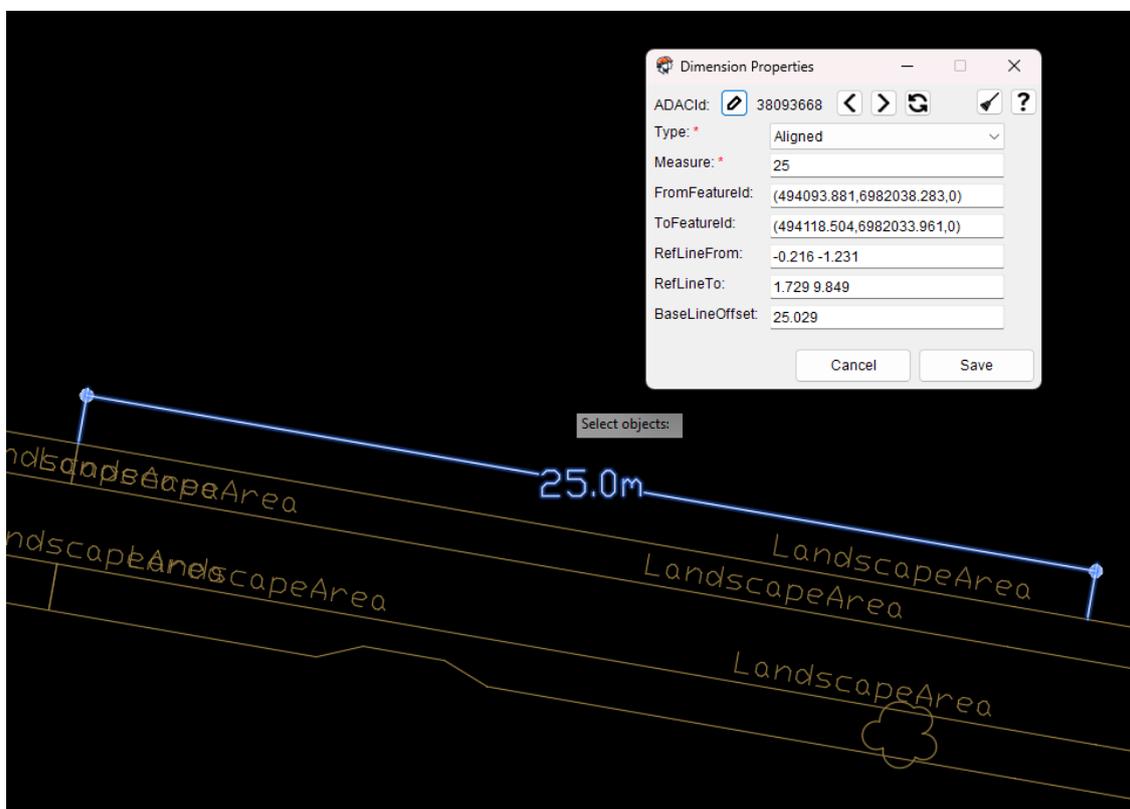
The form can be prefilled with extracted MText properties (for example text, font, height, rotation and justification) so these values can be confirmed and stored with the enhancement attributes.



Create a dimension enhancement

Use 'ADACDimension'  to assign an existing CAD dimension as a dimension enhancement and attach the required attributes.

- Select a single dimension first (optional). If one dimension is preselected, the command uses it automatically
- Run the 'ADACDimension'  command
- If no dimension is preselected, select the dimension when prompted
- Complete the attribute form and click OK to store the enhancement asset data on the dimension object



What is stored on the object

When an enhancement is created successfully, **infrADAC** writes asset metadata and user attribute values to XData on the CAD entity.

Core stored fields include:

- ID, Category, FeatureClass and Geometry
- ADACId (copied from global attributes)

Prefilled attribute values

Annotation prefill

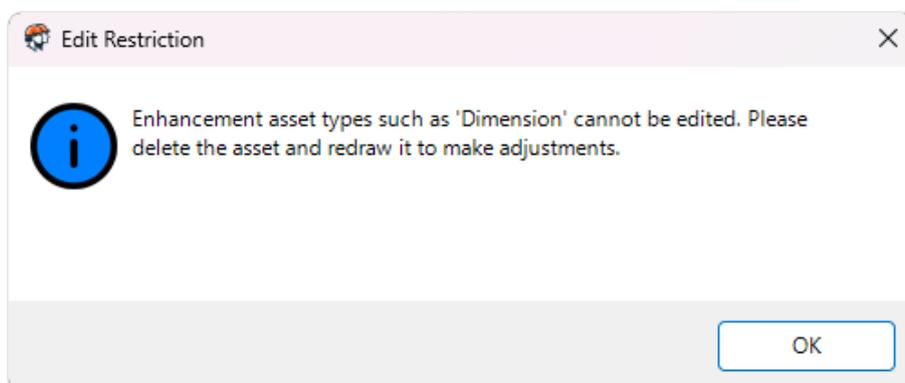
- Type (defaults to Warning in the current implementation)
- Text (plain text extracted from MText)
- Font (font name, text height, style and colour in hex format)
- Height_m
- Width_m (currently set to 0)
- Rotation (degrees)
- Justification

Dimension prefill

- Type (Aligned, Linear, Angular and so on, based on the dimension type)
- Measure (rounded to 3 decimals, angular measurements in degrees)
- FromFeatureId and ToFeatureId (point values used to derive the dimension)
- RefLineFrom and RefLineTo (reference line vectors used for display or reconstruction)
- BaselineOffset

Limitations and common issues

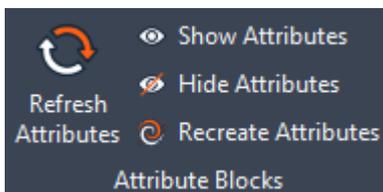
- Editing is not supported: delete and recreate the object to change stored attributes
- If enhancement definitions are missing from the JSON configuration, the command reports that no data is available
- If the selected object is not a valid dimension, the command exits with an invalid selection message
- If a dimension is already assigned as an asset, it cannot be assigned again



Handling Attribute Blocks

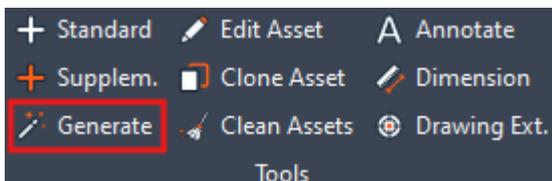
Attribute blocks are an informational feature of **infrADAC** and serve as a way for designers to quickly view the assigned attribute values without having to open any additional windows. They are only available for base ADAC XML assets but are not essential.

Users will find that larger designs with attribute blocks visible can quickly clog up the model space view, so it is suggested to keep these blocks visible only when needed. Utilising the **'ADACShowAttributes'**  and **'ADACHideAttributes'**  commands can toggle the view of the preselected assets or all assets if no preselection is made. **'ADACRefreshAttributes'**  is helpful to users who notice a discrepancy between the attribute block values and the values assigned to the asset object itself. The command will pull the attribute values from the asset and update the values displayed in the associated block. If for any reason the block was deleted, it can be recreated by calling the **'ADACRecreateAttributes'**  command.



Generate an Asset from existing object/s

Arguably one of the most useful features of **infrADAC** is the ability to create ADAC XML assets of all types of existing model space objects. By preselecting an object or objects (if 'multi' geometry is required) and running the **'ADACGenerate'**  command, **infrADAC** will open the **Asset Library** window and list the assets that can be created utilising the same geometry as the object/s selected. This ensures that on export, all model space objects that are assigned ADAC XML attributes will align to the expected geometry of their assigned asset. This feature extends to both standard and supplementary assets.



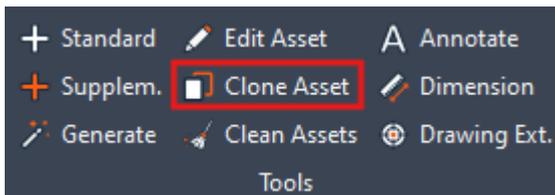
Cloning an Asset

Cloning an existing asset in the model space provides users with the ability to quickly and efficiently recreate assets that share the same attribute properties that may appear multiple times throughout the design. Assets such as trees, fittings, luminaires, pit and poles that are functionally identical don't have to be manually added each time they are required. The clone command **'ADACClone'**  will ensure that each asset defined is still assigned its own ADACId (ObjectId if working in v4.1). The preselected object must already be an ADAC XML asset so at least one must be defined before you can clone for the others.

Cloning lets you quickly create copies of an existing asset that already has the correct attributes, feature class and settings. This is ideal for repeating items such as trees, pits, poles, fittings and luminaires. Each cloned asset is still given its own unique ADACId so it remains a separate asset for audit and export.

'ADACClone'  has two different cloning actions, depending on whether you are placing clones manually or cloning to preselected target positions.





Before you start

To use 'ADACClone'  you must already have at least one valid ADAC asset in the drawing to clone from. The source object must be a real **infrADAC** asset, not just a similar looking block or linework. Annotation and Dimension assets cannot be cloned with 'ADACClone' .

Cloning Method 1: Place clones by clicking insertion points

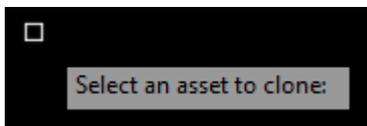
Use this when you want to place one or more clones interactively.

How to do it:

- Select one existing asset in model space (this will be your source asset).
- Run 'ADACClone' .
- Click an insertion point in the drawing for each clone you want to place.
- Press Enter to finish placing clones.
- A confirmation message appears showing how many clones were created.

What you should expect

After each click, a new asset appears at the point you chose. The clone carries the same attributes as the source asset, but with a new ADACId.



Cloning Method 2: Clone to preselected target positions

Use this when you already know the target locations and you want cloning to happen in one step.

This action is designed for workflows where you preselect multiple objects that represent the target positions. These can be points, blocks or other objects with a clear location.

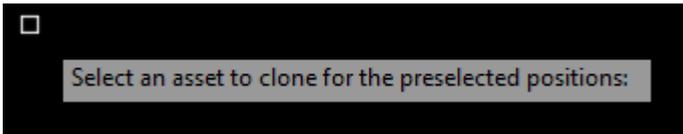
How to do it

1. In model space, preselect the objects that represent your target positions.
Note: You must select more than one object for this mode to activate.
2. Run 'ADACClone' .
3. When prompted, select the source asset you want to clone.
4. **infrADAC** creates a clone at each preselected target position.
5. A confirmation message appears listing the created ADACIds.

What you should expect

You will not be asked to click insertion points. Clones are created automatically at the locations of the objects you preselected.





Important: Target positions are taken from the location of the preselected objects.

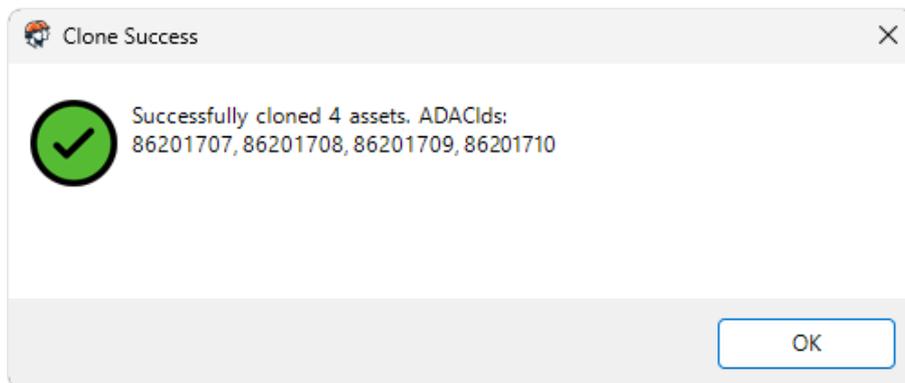
Most commonly this will be:

- Points
- Blocks
- Simple objects where a centre or insertion location is obvious

If a selected object does not provide a usable location it is skipped.

What happens to IDs

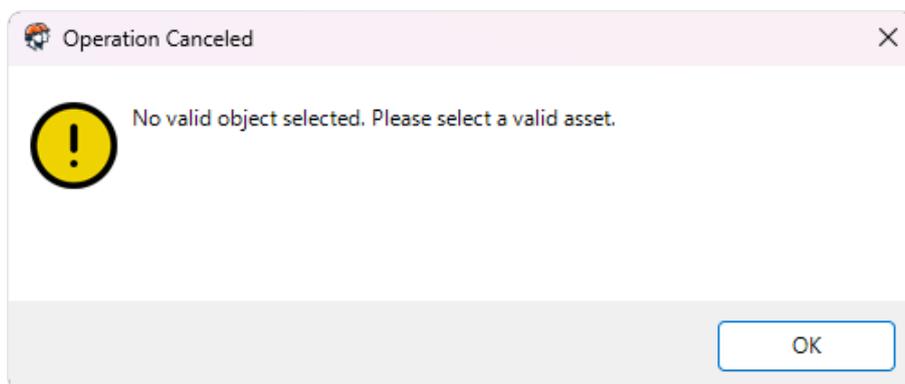
When a clone is created it is assigned a new ADACId automatically. The original asset keeps its existing ADACId. In many cases the new IDs will increment in a sensible sequence for that asset type, otherwise a new numeric ID is created. You do not need to type or manage IDs during cloning.



Common issues

“No valid object selected.”

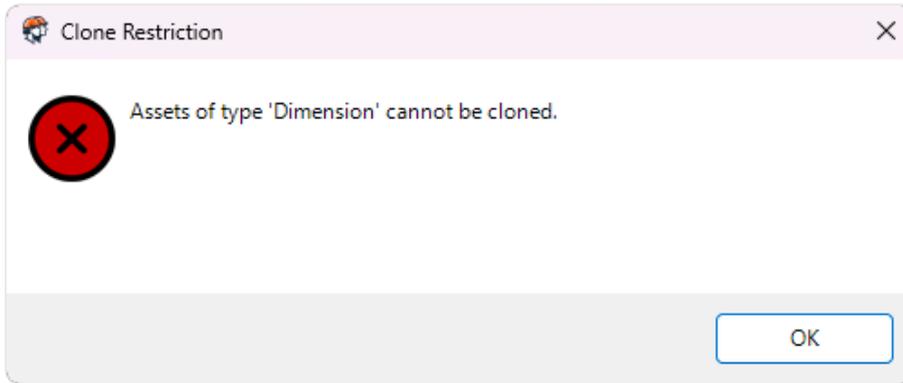
You selected something that is not an **infrADAC** asset. Make sure the source object was created through **infrADAC** and has asset data.



“Assets of type ‘Annotation’ or ‘Dimension’ cannot be cloned”

These types are intentionally blocked. Create new annotations or dimensions using the normal enhancement tools instead of cloning.





Cleaning Assets

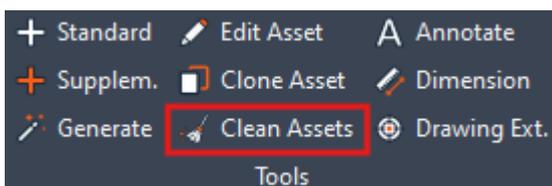
When users wish to ensure that fundamental attributes of all assets within a design are accurate and match the expected requirements within the profile, the **'ADACClean'**  command can be called at any time before audit and export. This is a command that should not be necessary to run but can be used as a point of last resort to ensure the design won't cause any obvious issues. The following actions are performed when the command is run:

- ADACId/ObjectId attribute is regenerated.
- Model space objects that act as a single multi type geometry asset are regrouped.
- Attribute blocks that are missing are recreated.

Important: If you enable the option to renumber ADACIds, both manually entered and automatically created IDs can be cleared and replaced. Only run this if you understand that IDs may be reassigned.

When users want to make sure the core information stored on assets is consistent and reliable before running an Audit or Export the **'ADACClean'**  command can be used. This is a maintenance tool intended as a "last resort" if a drawing has been heavily edited, copied between files or is behaving unexpectedly during audit. In most normal workflows it should not be required.

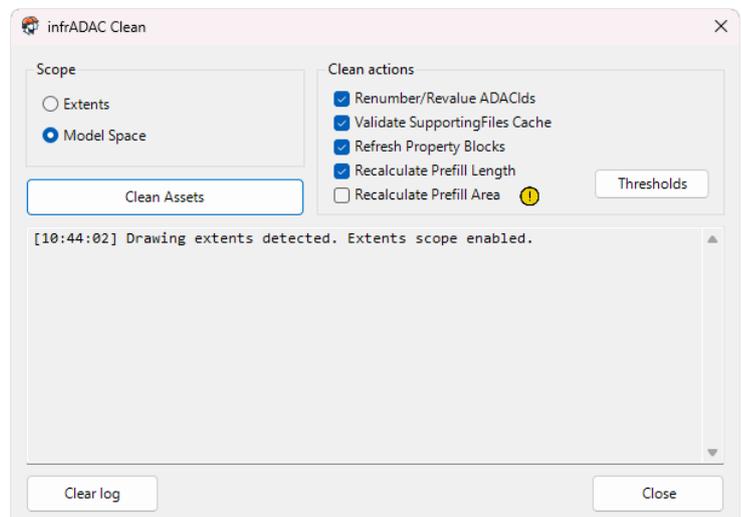
Running **'ADACClean'**  opens the **infrADAC Clean** window where you choose what to clean and whether to apply it to the whole model space or only within your saved drawing extents (if extents exist for the project).



Where to run ADACClean

Open the AutoCAD/BricsCAD command line and run: **'ADACClean'** 

The Clean window opens as a modeless dialog so you can leave it open while you work.



Understanding Scope

'ADAClean'  can work on either:

- **Extents:** Only assets that fall inside the project DrawingExtents (only available if valid DrawingExtents exist in Project Attributes)
- **Model Space:** All detected assets in the current drawing

If Extents are not available, the Extents option is **disabled** and Model Space is used automatically.

Clean Actions

Renumber/Revalue ADACIds

This option clears existing ADACId values for assets in scope then assigns new unique ADACIds.

What the user should know before selecting it:

- Existing IDs may be replaced even if they were manually entered
IDs are reassigned to remove duplicates and fix conflicts
- Assets with a recognised prefix (an "alias") keep that prefix and get new sequential numbers after it where possible
- Assets without a prefix are assigned an 8 digit numeric ID starting from a random point then incrementing to avoid clashes

After renumbering, **infrADAC** also attempts to keep SupportingFiles consistent by updating folder names and stored paths if an asset's ADACId changed.

Validate SupportingFiles Cache

This option checks the temporary SupportingFiles cache used by **infrADAC** for the current project.

Resulting action:

- Check for missing supporting files referenced by assets.
- Remove orphan files from the cache that are no longer referenced by any asset.
- Remove empty folders left behind after cleanup

This does not delete the user's original source files. It only cleans the **infrADAC** temp cache.

Refresh Property Blocks

This option repairs the property display blocks used by **infrADAC**.

Resulting action:

- Recreate missing asset property blocks where needed.
- Refresh all asset property blocks so the displayed information matches the latest stored attributes.

Use this if property panels are missing, duplicated or not reflecting recent edits.

Recalculate Prefill Length

This option recalculates length based values from the actual geometry and updates the stored "prefill" attributes only when the difference is large enough to matter.

It is intended for cases where length attributes are out of sync with edited geometry.



Recalculate Prefill Area

This option recalculates area based values from the actual geometry and updates stored “prefill” area attributes only when the difference is large enough.

Important: This can overwrite area values that were intentionally set by a user. Only enable this if area values should always match geometry.

Thresholds (Prefill recalculation)

If you select Recalculate Prefill Length or Recalculate Prefill Area the Thresholds button becomes available. Thresholds control how big the difference must be before **infrADAC** overwrites the stored value. This prevents small rounding changes from constantly rewriting attributes.

Use Thresholds if:

- Values are being updated too aggressively.
- Values are not being updated when you expect them to be

How to run a clean

1. Run ‘ADACclean’  from the command line.
2. In Scope choose Extents or Model Space.
3. Tick the clean actions you want to apply.
4. Click Clean Assets.
5. Review the log output in the window.
6. When complete review the completion message and any warnings.

What you will see during and after cleaning

The log panel shows progress messages with timestamps, including how many assets were detected and what actions were performed. When cleaning finishes a summary message appears showing what was updated and listing warnings if anything could not be cleaned.

Common issues

- If Extents is greyed out: the project does not currently have valid DrawingExtents saved in Project Attributes so cleaning must run across Model Space.
- If you see warnings about missing supporting files: one or more SupportingFiles paths stored on assets point to files that no longer exist in the cache or were moved outside it.
- If IDs changed unexpectedly: you likely ran ReNUMBER/Revalue ADACIDs. This is expected behaviour and is the reason the tool is considered a last resort.



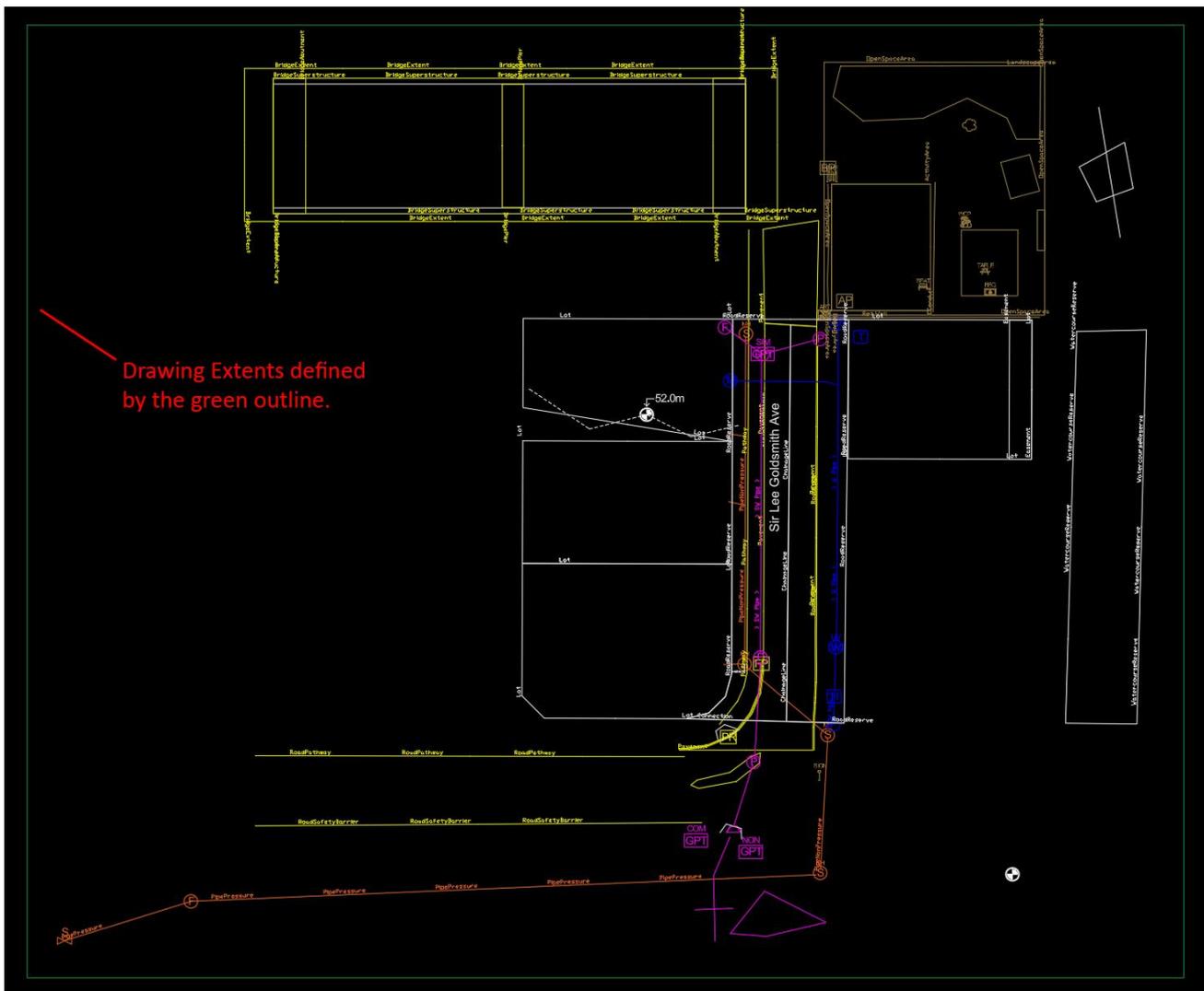
Drawing Extents

Drawing extents define a rectangular boundary using two XY coordinate pairs:

- SouthWest (X,Y): the lower left corner
- NorthEast (X,Y): the upper right corner

In **infrADAC** you will interact with drawing extents in two separate actions:

- **Setting the drawing extent limits** stored with a profile (this controls what the profile considers valid limits)
- **Setting the drawing extents** for the current design (this stores extents against the design via Project Attributes)



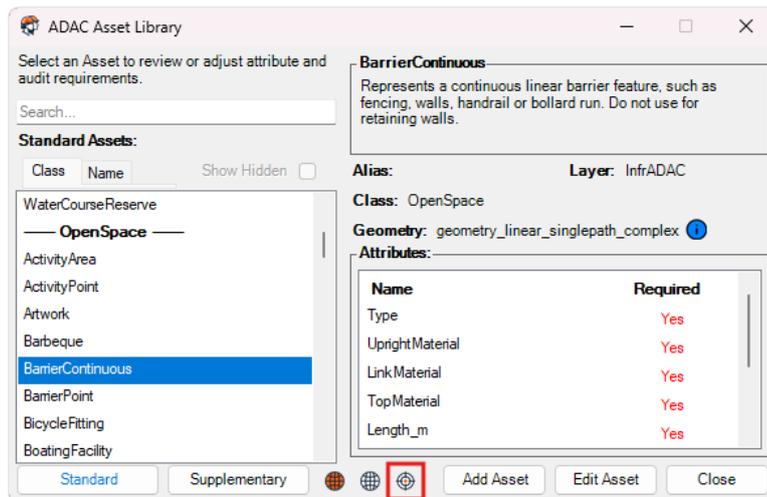
Setting the Drawing Extent Limits

Use this action when you want to define or adjust the drawing extent limits that are saved with your profile settings.

Where to find it

Open the Asset Library and locate the Drawing Extents button in the bottom toolbar. The tooltip reads: Set drawing extent limits. Click it to open the Drawing Extents window.

If the button is unavailable, the profile is not loaded (embedded mode). In this state the program blocks editing and shows a message that a profile must be loaded.

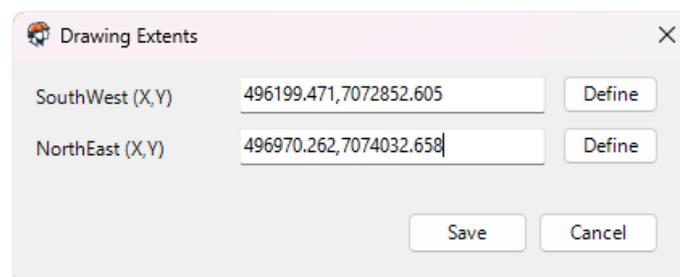


Drawing Extents window

- The window contains:
- SouthWest (X,Y) field and a Define button
- NorthEast (X,Y) field and a Define button
- Save and Cancel

How to set or edit profile limits

1. Open the Asset Library.
2. Click the Drawing Extents button (Set drawing extent limits).
3. Set SouthWest and NorthEast using one of the options below.
 - Pick points from the drawing
 1. Click Define next to SouthWest then pick the SouthWest corner in the drawing.
 2. Click Define next to NorthEast then pick the NorthEast corner in the drawing.
 - Type coordinates manually
 1. Enter SouthWest (X,Y).
 2. Enter NorthEast (X,Y).
4. Click Save.



Coordinate format for manual entry (profile limits)

Manual entry must contain exactly two numbers and supports separators:

- X,Y
- X Y
- X;Y

Common issues

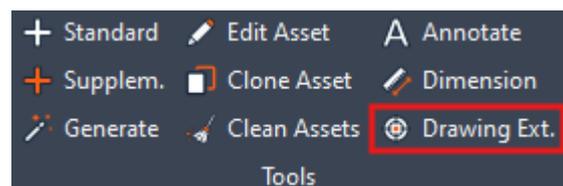
- If you cannot click the Drawing Extents button: a profile must be loaded, embedded mode blocks editing.
- If Save says coordinates are invalid: ensure each field has exactly two numbers and uses dot decimals.
- If values seem reversed: SouthWest must be lower left and NorthEast must be upper right.

Setting the Drawing Extents for the current design

Use this action when you want to set the drawing extents stored with the current design through Project Attributes. This is the extents users are setting for the design, not the profile limits.

Where to find it

- Open the Project Attributes window.
- Scroll to the DrawingExtents section.
- Click the target icon next to the DrawingExtents label. The tooltip reads: Set Drawing Extents
- Use the command **'ADACDrawingExtents'**  .



How to set design drawing extents using the target icon

1. Click the target icon in the DrawingExtents section.
2. The Project Attributes form temporarily hides while you interact with BricsCAD.
3. Follow the prompts in AutoCAD/BricsCAD to define, confirm, regenerate or redraw the extents rectangle as required.
4. When complete, the Project Attributes form returns and automatically fills:
 - SouthWest
 - NorthEast
5. Click Save in Project Attributes to store the updated drawing extents for the design.

DrawingExtents: 	
SouthWest: *	493780.513,6981943.147
NorthEast: *	494199.013,6982451.714

Manual entry rules in Project Attributes (design extents)

You can type coordinates directly into SouthWest and NorthEast. These fields require:

Exactly two numeric values separated by a comma

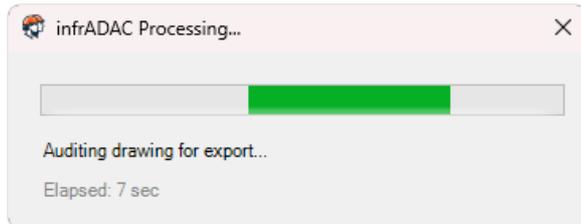
Example: 123456.789, 987654.321

Important: When manually specifying the extents allowed characters are digits, decimal point, minus sign, comma and spaces. Alphabetic characters are not allowed.



Producing an Audit Report

A successful audit report must be achieved before a design can be exported to an ADAC XML file for submission. The audit report can then be used alongside the ADAC XML file as proof of achieving the specified standard, as well as providing a comprehensive breakdown of the project attributes and individual asset attributes. Depending on the specifications and requirements of the loaded profile, assets are audited based on their allowed type, required attributes, specified enumeration values, geometry, feature element etc; even specific ComponentInfo Notes requirements are audited.



Before the audit starts

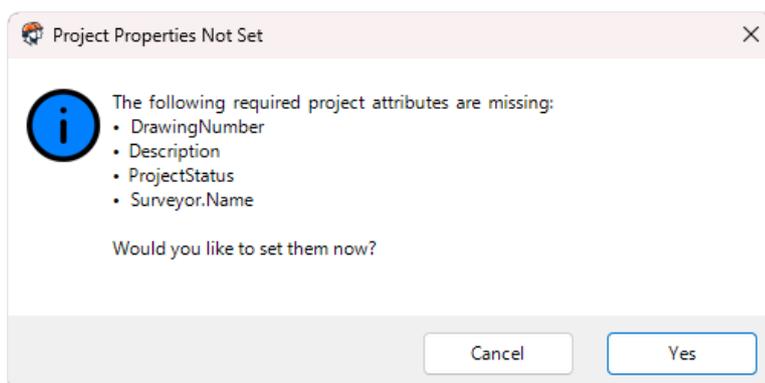
The audit command '**ADACAudit**'  first performs the normal licence and disclaimer checks. It also requires the system to have already been initialised. If the project has not been initialised, the audit does not continue and the user is told to initialise the system with '**ADACNew**'  first.

Before opening the audit window, **infrADAC** checks that the required **project attributes** are present. This uses the same project data requirements as the export workflow.

If required project attributes are missing, the user is shown a prompt listing the missing fields and asked whether they want to set them now.

- **Yes** - Opens the project attributes form so the missing required values can be entered.
- **Cancel** - Stops the audit.

If the project attributes form is cancelled or does not complete successfully, the audit is also cancelled.



Audit window layout

The audit window displays the audit parameters at the top and the live summary results underneath.

The visible controls in the audit area are:

- **Profile** - Selects the profile used for the audit.



- **Highlight Failed Assets** - Controls whether failed assets are highlighted in the drawing.
- **Full Technical Report** - Indicates whether the full report output is available.
- **Convert Complex Geometry** - Used when complex curved geometry is detected and conversion is permitted.

Below these controls, the audit summary section shows:

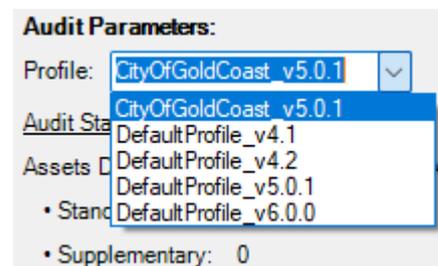
- **Audit Status**
- **Assets Detected**
- **Standard**
- **Supplementary**
- **Assets Passed**
- **Standard Passed**
- **Supplementary Passed**

Profile selection

The **Profile** drop-down controls which extracted profile is used for the audit. When the selected profile changes, **infrADAC** reloads the profile into a temporary audit location, refreshes the profile metadata and reruns the audit against the current model space design.

This means the user can immediately compare the same drawing against a different installed profile without leaving the audit window.

If the selected profile cannot be read correctly, the audit will fail against that profile configuration.



What the audit checks

The audit examines both project-level data and asset-level data.

Project-level validation

The audit validates the project data against the active profile, including:

- required project attributes
- selectable option values
- hardcoded expected values
- drawing extents values
- coordinate system related requirements where applicable

A project-level issue can cause the overall audit to fail even if all detected assets pass.

Asset detection in model space

The audit scans model space and builds the asset counts from the detected **infrADAC** content. It separates the results into:

- total assets detected
- standard assets
- supplementary assets
- grouped assets



- single assets

The displayed counts are then used to populate the summary labels and the technical report.

Asset validation

For each detected asset, the audit validates:

- whether the asset exists in the selected profile
- whether the geometry type matches the expected profile geometry
- whether the required global attributes are present
- whether the required asset attributes are present
- whether selectable option values are valid
- whether hardcoded values are correct
- whether grouped assets are complete and internally consistent
- whether ADACId/ObjectId values are duplicated

If an asset fails one or more checks, it is counted as failed and can be highlighted in the drawing.

Grouped assets

Grouped assets are audited as a single logical asset rather than as separate unrelated entities. During the audit, **infrADAC** checks that:

- all expected grouped entities are present
- the group GUID is consistent
- grouped attribute values are consistent where required
- the expected group count is satisfied

If a grouped asset is incomplete or inconsistent, the group fails and the affected group entities are included in the failed highlight set.

Geometry validation

The audit compares each asset against the geometry expected by the active profile.

Examples of geometry issues that can cause a failure include:

- the wrong geometry category for the asset
- geometry complexity that exceeds the expected limitation
- grouped or multi-geometry content where only single geometry is allowed
- annotation or dimension entities that do not match the required entity type

When geometry does not match the profile requirement, the asset is marked as failed.

Duplicate ADACId/ObjectId detection

After the assets have been processed, the audit checks whether the same **ADACId/ObjectId** has been used more than once across separate assets or groups.

If a duplicate ADACId/ObjectId is detected:

- each affected asset is marked as failed
- the ADACId/ObjectId failure is recorded in the report
- the duplicate issue is added to the overall audit issue list



Audit status display

The **Audit Status** field shows the final overall result of the audit.

The status can display:

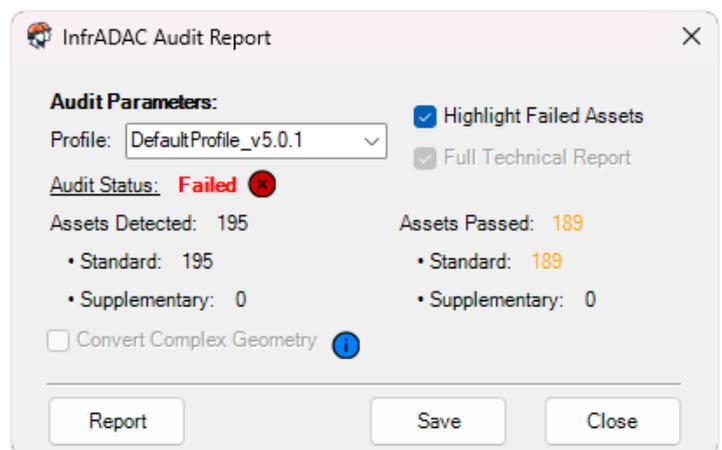
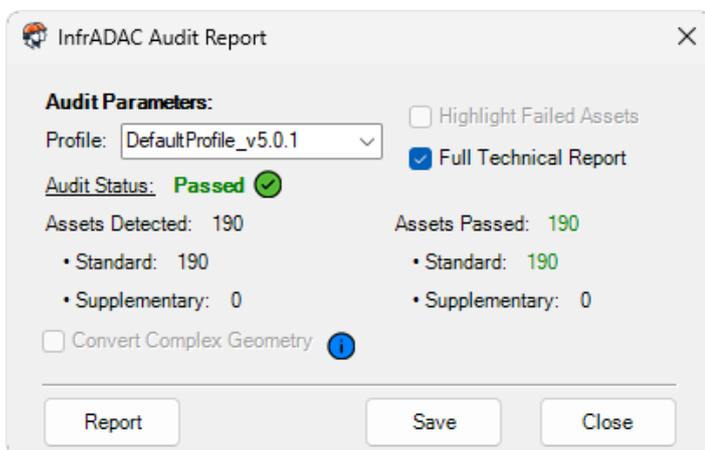
- **Passed** - All required project checks and asset checks have passed.
- **Failed** - One or more required project checks or asset checks have failed.

The status is colour-coded and paired with a status icon.

- **Green** indicates a successful audit
- **Orange** indicates a warning-style result where asset results may be acceptable but project-level issues still remain
- **Red** indicates a failed audit



Hover help on the status icon shows the recorded audit issues in a readable list.



Asset count and pass count summary

The summary labels show the audit totals for the current run.

- **Assets Detected** - Shows the total audited assets found in model space.
- **Standard** - Shows the count of standard assets.
- **Supplementary** - Shows the count of supplementary assets.
- **Assets Passed** - Shows how many of the detected assets passed the audit.
- **Standard Passed** - Shows how many standard assets passed.
- **Supplementary Passed** - Shows how many supplementary assets passed.

The pass-count labels are colour-coded to reflect whether all, some or none of the assets in that category passed.

Tooltips on the count labels provide a grouped-versus-single breakdown.



Full Technical Report

The audit process generates a Word based technical report using the current audit result set. This report is the main user-facing output of the audit.

The report includes:

- overall audit result
- active profile used for the audit
- project attribute values
- summary counts for standard, supplementary, grouped and single assets
- asset-by-asset audit results
- failed fields and validation reasons
- generated ADAC XML content when the audit passes

Failed values and failed attributes are marked visually in the report so the problem areas are easy to identify.

The **Full Technical Report** option is enabled only when the report output is available for use.

Report generation behaviour

When the audit completes, **infrADAC** generates the audit report into its temporary export location and then shows a result prompt.

If the audit passes, the user is told that the audit was successful and can choose to open the report.

If the audit fails, the user is told how many issues were detected and can still open the report to review the details.

The result prompt provides:

- **Report** - Opens the generated audit report.
- **Continue** - Returns without opening the report.

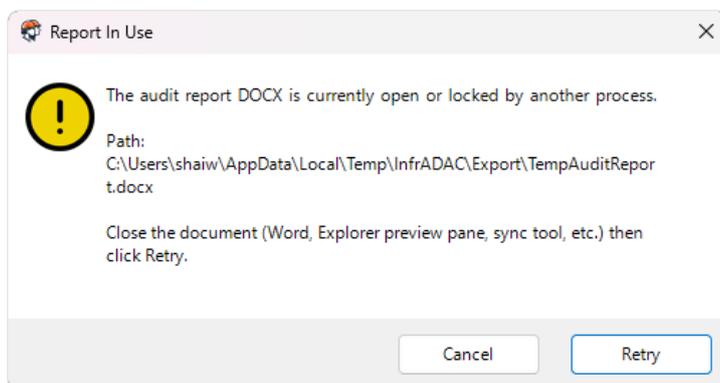
Report file in use behaviour

If the report file is already open, **infrADAC** detects that the file is in use and shows an appropriate message.

Depending on the situation, the user may be offered:

- **Retry** - After closing the open report file
- **Cancel** - To stop the action

This prevents the existing report from being overwritten while it is still open in Word.



Convert Complex Geometry

The audit also detects whether complex curved geometry exists in the current design.

If complex geometry is found, the control records that information so it can be used later by the export workflow. The **Convert Complex Geometry** option only becomes available when all of the following are true:

- the workflow supports export conversion
- the audit has passed
- complex curved geometry has been detected



If these conditions are not met, the option remains disabled and its tooltip explains why it is unavailable.

For a normal audit-only review, this option may remain unavailable unless the audit is being used as part of an export-capable workflow.

Audit outcomes

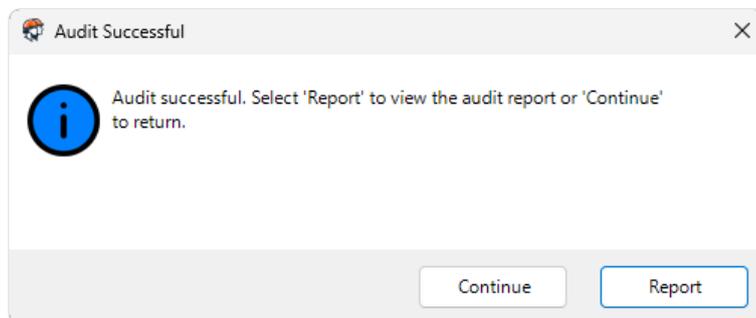
The audit can finish in two broad ways.

Passed audit

A passed audit means:

- required project attributes are valid
- detected assets passed validation
- no blocking geometry or profile compliance issues remain

In this state, the report records a successful outcome and the generated ADAC XML content is available within the report output.



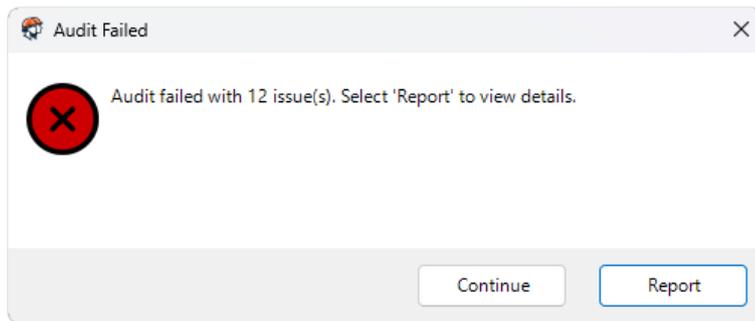
Failed audit

A failed audit means one or more problems were detected, such as:

- missing required project attributes
- invalid project values
- asset profile mismatches
- geometry mismatches
- missing required attributes
- invalid selectable option values
- duplicate ADACId/ObjectId values
- grouped asset consistency failures

In this state, the report is still generated so the issues can be reviewed in detail.





Typical uses

The Audit Report function is used to:

- check whether the current design is ready for export
- verify that project data has been completed correctly
- confirm that assets match the selected profile
- find failed assets in model space quickly
- generate a technical record of the audit result for review or issue resolution

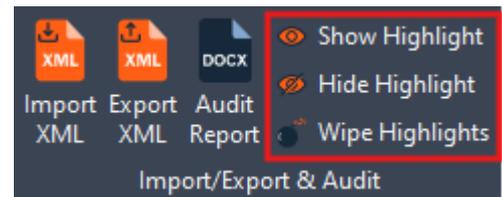
Tips

- The audit works against the **current model space design**.
- Project attributes may need to be completed before the audit window opens.
- Changing the profile reruns the audit against the same drawing content.
- Failed assets can be highlighted in the drawing for easier review.
- The technical report is generated even when the audit fails so the issues can be reviewed.
- If the report is already open, **infrADAC** will prompt before replacing it.

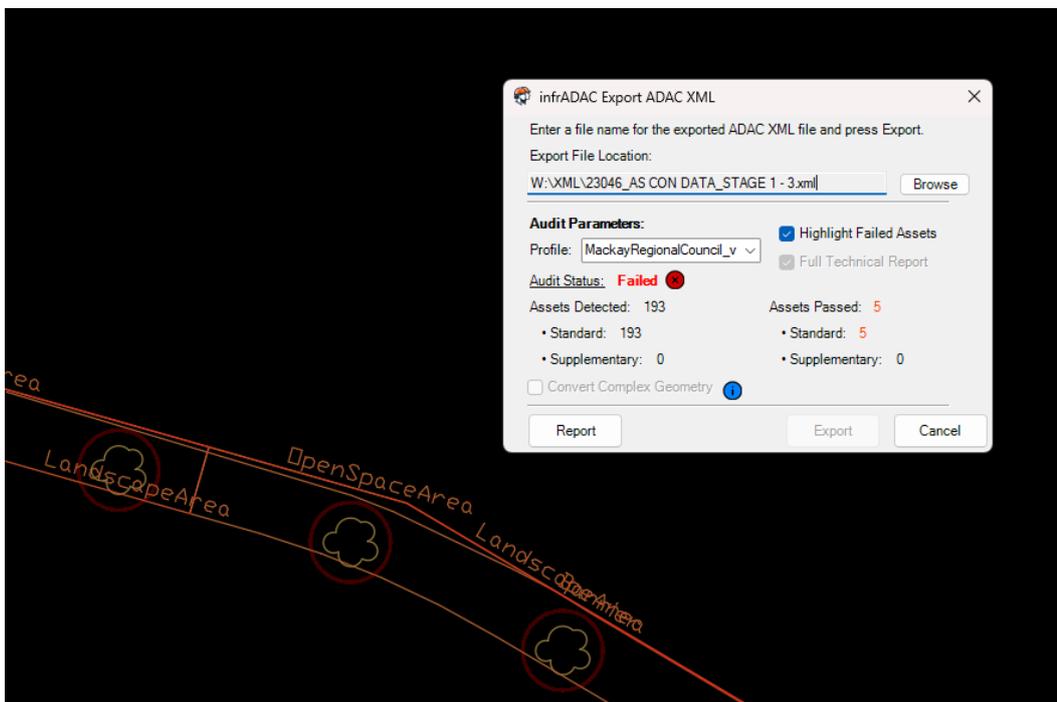


Audit Report Error Highlights

When the audit identifies an issue within the design that is to be exported, the specific asset object/s in the model space are highlighted to assist the user in quickly identifying which assets are conflicting with the expectations of the loaded profile. If the user has not enabled line thickness, either by unchecking the option for the ADAC New actions in the **Settings Window** or within AutoCAD's own settings, error highlights will not be visible.

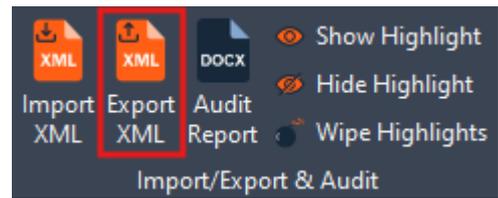


Users may show error highlights in the model space by calling the **'ADACErrorShow'**  command or hide them with the **'ADACErrorHide'**  command. Sometimes error highlight can linger from deleted assets or assets that are fixed can be cleared using the **'ADACErrorWipe'**  command.



Exporting an infrADAC design to ADAC XML

The Export feature creates an ADAC XML file from the drawing you currently have open in AutoCAD/BricsCAD. As part of the export process, **infrADAC** also produces an audit report so you can confirm the drawing meets the required checks before you send the XML to someone else or upload it into another system.



When you open the Export window, **infrADAC** runs an audit automatically. The Export button stays unavailable until you choose where to save the XML and the audit has passed.

Starting the export

1. Open the drawing you want to export and make sure it is the active drawing.
2. Open the Export window called “**infrADAC** Export ADAC XML”.
3. Wait while the audit runs. A small processing message may appear while the audit is running.

Once the audit completes, the Export window shows the file location area, the audit results and the buttons along the bottom of the window.

Choosing where the XML will be saved

1. Select Browse.
2. Choose the folder where you want the export saved.
3. Type a file name for the XML.
4. Select Save.

The file will be saved with the .xml extension. If you type a name without an extension, **infrADAC** will still save it as an XML file.

Reviewing the audit results

The audit section explains whether your drawing is ready to export. If the audit does not pass, the Export button remains disabled. In that case, review the issues shown in the audit section, return to the drawing to correct them and then open the Export window again to re-check.

If you want to read the audit results in a full report, use the Report button after the report has been generated.

Exporting the XML

1. Confirm the save location shown in the Export window is correct.
2. Confirm the audit has passed and the Export button is enabled.
3. Select Export.

A progress message may appear while the files are being written. When the export completes, **infrADAC** shows a confirmation message that lists the files created and where they were saved. The confirmation message may also include a quick option to open the report or open the export folder.

What files you get after exporting

Export produces an ADAC XML file at the location you selected. It also produces a Word audit report saved in the same folder using the same base file name, with “_AuditReport” added to the end of the name. In some



projects, export can also include a ZIP file of supporting files. That ZIP is only created when supporting files exist and can be packaged successfully.

Common issues

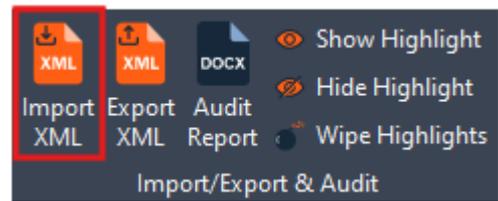
If you see a message saying a file is in use, it usually means the report document or ZIP file is already open in Word or being previewed by File Explorer. Close anything that might be using the file and retry the export.

- If the Report button is disabled, it means the report has not been generated yet. Run the audit by opening the Export window and wait for the audit to complete, then the report option becomes available.
- If the Export button is disabled, it means either the audit has not passed or you have not selected an export location yet. Choose a save location first, then resolve any audit issues until it passes.



Importing an ADAC XML file into model space

ADAC XML files that are received for review can also be audited and imported into the CAD model space. Regardless of where the file originates or which software created the file, **infrADAC** allows for an audit to be performed before import. An important note is that the file must pass the audit process before it can be imported to the model space. This ensures the assets that are added will not conflict with the loaded profile it was audited against.



Before you begin

Before starting the import, make sure the following conditions are met:

- the correct drawing is open
- the ADAC XML file has already been reviewed using the process described in the **Audit** section
- the file to be imported is the final version intended for use
- you have permission to modify the current drawing

The import process places content into the active drawing, so it should only be run in the file that is intended to receive the ADAC data.

Opening the import form

To begin, open the **Import ADAC XML** form from **infrADAC**.

The form is used to select the XML file and start the import. It includes a file location field, a **Browse** button and an **Import** button. The **Import** button is only available once a valid file has been selected and the file has been assessed as suitable for import.

Selecting the ADAC XML file

Click **Browse** and navigate to the required ADAC XML file.

Once selected, the full file path is stored and displayed in the form. If the path is too long to fit in the field, it may appear shortened in the text box, but the full path remains associated with the selected file.

After the file is selected, **infrADAC** reads the XML and prepares it for import. This includes checking the file content and preparing the data that will be used to create drawing objects.

Starting the import

To start the import, click **Import**.

A processing window is displayed while the file is being imported. This window remains open until the import process has finished. During this time, **infrADAC** reads the ADAC XML data and creates the corresponding objects in model space.

The drawing should not be edited until the import process is complete.

What is imported into model space

The import process creates model space objects from the data contained in the ADAC XML file. Depending on the contents of the XML, this may include standard assets, supplementary features and enhancement objects.

In practical terms, the imported content may include:



- point based assets
- line and polyline based assets
- polygon based assets
- annotation objects
- dimension objects
- the drawing extents boundary from the XML file

Each imported object is positioned according to the coordinates stored in the ADAC XML file. This means the imported content is created at its intended location rather than being placed manually by the user.

How imported objects are created

When the XML is imported, **infrADAC** converts the ADAC data into drawing objects that can be displayed and managed in the current drawing.

Standard asset geometry is created in model space and then processed as **infrADAC** assets so that the imported objects carry the information needed for ongoing use. Where applicable, imported objects are assigned the attribute data associated with the XML record.

If the XML includes annotation or dimension style content, those items are also created as drawing objects so they appear visually in model space as part of the imported dataset.

Where drawing extents are present in the XML file, **infrADAC** also creates an extents boundary in model space. This gives a visible reference for the imported area and helps show the overall footprint of the imported data.

Import behaviour for identifiers

Each imported asset is expected to have an ADAC identifier. Where the XML contains a valid identifier that does not conflict with existing imported data, that identifier is retained.

If an identifier is missing, invalid or duplicates an existing identifier, **infrADAC** may assign a replacement identifier so the asset can still be created successfully. This allows the import to proceed while maintaining unique identifiers for imported objects.

This behaviour helps prevent the import from failing simply because one or more identifiers in the XML are unsuitable for direct use.

Import behaviour for geometry

Imported geometry is created according to the type of geometry contained in the XML file. This can include single point geometry, multi point geometry, linear geometry and area geometry.

Where grouped geometry is present, **infrADAC** attempts to recreate that grouped structure in model space using the appropriate drawing objects. Circular and elliptical geometry is also interpreted where supported by the XML content.

If geometry cannot be created correctly for a particular asset, that asset may be skipped or counted as failed in the final import summary.

What happens after import

When the import has completed, **infrADAC** updates the drawing view to the imported area. This makes it easier to immediately see the newly created content in model space without manually zooming to find it.



The drawing is then regenerated so the imported objects display correctly.

A completion message is shown at the end of the process. This summary reports:

- how many assets were created
- how many assets were skipped
- how many assets failed to import
- how many assets were parsed from the file
- whether duplicate ADAC ID groups were present
- how many assets were outside the drawing extents

This summary gives the user a clear indication of the overall import result.

If some items are not imported

Not every XML item is guaranteed to import successfully. An item may be skipped or fail if its geometry cannot be interpreted, if the required data is incomplete or if the content cannot be created correctly in the drawing.

A partial import can still complete successfully overall. In this situation, the summary message should be reviewed so the number of skipped or failed assets is understood.

If the import result is not as expected, return to the **Audit** section and review the XML before attempting the import again.

Good practice

When importing an ADAC XML file into model space, it is good practice to confirm that:

- the imported objects appear in the expected location
- the visible extent of the imported data matches the expected project area
- annotation and dimensions appear correctly where included
- the number of created assets is reasonable for the file being imported
- any skipped or failed items are investigated before relying on the imported result

This review should be carried out immediately after the import so that any issues are identified early.

infrADAC Log

'ADACLog'  opens the latest infrADAC log file. This gives the user quick access to the most recent logging information without having to browse to the log folder manually.

The command is used when recent system activity, warnings, or errors need to be reviewed. It is intended as a simple support and troubleshooting command.

When ADACLog runs infrADAC:

- searches the log folder for the most recent log file
- opens that file using the default Windows application for .log files
- shows a message if no log file is available
- shows an error message if the file cannot be opened



Updating a Licence

The licence update process is used to replace the currently installed licence.dat file with a newer valid licence. This may be required when activating a new licence, renewing an expiring licence or replacing an invalid licence with a corrected one.

The process begins from the **Update Licence** button in the Settings window. From there, the user is prompted to choose whether the update should come from the server or from a local licence file.

Available update methods

Two update paths are supported.

- **Server update** - Downloads the latest licence from the configured server endpoint, validates it and installs it automatically if successful.
- **Local file update** - Allows the user to browse to a licence.dat file supplied outside the application, validate it and install it as the active licence.

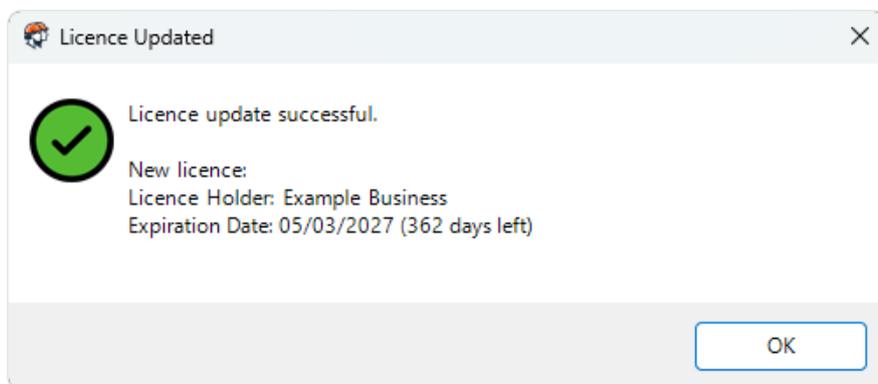
Both methods include validation checks before the new licence is accepted.

Server update

The server update path is the recommended method when online access is available and the licence service is reachable. When selected, **infrADAC** attempts to contact the configured licence endpoint, download the updated licence and replace the installed licence automatically.

If the downloaded licence is valid, the licence details displayed in the Settings window are refreshed immediately. This includes the activation status, expiry date and registered business name.

If the downloaded licence fails validation, **infrADAC** does not install it and the existing state remains unchanged.



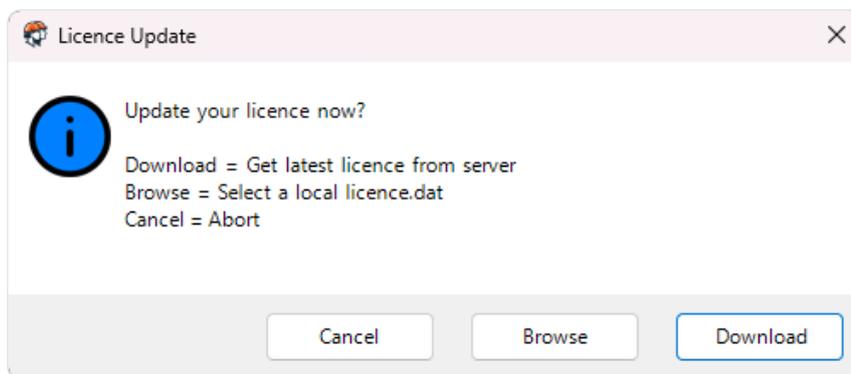
Local file update

The local file update path is used when a licence file has been supplied directly and needs to be installed manually. The user is prompted to browse to a licence.dat file and select it for validation.

Before installing a locally selected file, **infrADAC** performs checks to prevent meaningless replacements. It blocks the following cases:

- selecting the currently installed licence file by path
- selecting a licence file that matches the currently installed licence content
- selecting a licence file that does not advance the required expiry state in cases where a refreshed licence is specifically required

If the selected file passes validation, **infrADAC** installs it to the standard licence location and reloads the licence information shown in Settings.



Validation and failure behaviour

If the selected or downloaded licence cannot be validated, **infrADAC** does not install it. In this case the licence state remains invalid, expired or unchanged depending on the starting condition.

Typical invalid cases include:

- unreadable licence file content
- failed verification
- invalid or expired licence data
- downloaded or selected file not matching required integrity or licence rules

Where validation fails, **infrADAC** updates the displayed status accordingly and shows an error message.

After a successful licence update

After a successful update, the Settings window refreshes the visible licence information. This allows the user to confirm immediately that the correct licence has been applied.

The updated details normally include:

- activation status
- expiry date
- registered company name

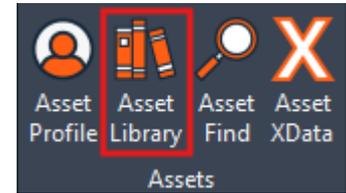
If the previous state had disabled most settings controls, restoring a valid licence allows normal access to the disabled Settings and profile controls again.



Windows and Forms:

Asset Library

The Asset Library window is the main browser for **infrADAC** asset definitions. It is used to search, sort and review assets, inspect their details and select an asset for Create or Generate workflows. When the window is opened in library mode, it also serves as the central location for accessing asset library management tools. Users can open the Asset Library window by selecting it in the top ribbon bar tab or by calling the '**ADACLibrary**'  command.



The window is split into two main areas. The left side is used to browse and filter the available assets. The right side displays the details of the currently selected asset, including its description, alias, layer, class, geometry and attribute requirements. This layout allows the user to quickly confirm whether the selected asset is the correct one before proceeding.

Required status legend

The Required column in the attributes list uses the following status values and indicators:

- **Yes** - The attribute is required by the loaded profile and the current asset definition.
- **No** - The attribute is not required by the loaded profile and current asset definition.
- **Yes*** - The attribute is marked as required in the loaded profile, but this requirement sits outside the embedded ADAC standard comparison for the current standard version.
- **No**  - Where the loaded asset requirement does not match the embedded ADAC standard expectation, the difference is shown with a caution icon and an explanatory tooltip note.

What the Asset Library window is used for

The Asset Library window is designed to support three main activities. Which controls are visible depends on how the window was opened, but the browsing and review experience remains broadly the same.

Create selection workflow

When opened as an asset picker for creation, the window displays a **Create** button. The user selects an asset from the list and confirms that selection. Assets that are rejected for the current context cannot be selected.

Generate selection workflow

When opened for generation, the window displays a **Generate** button and may also show a **Show Hidden** option. In this mode, rejected assets can be displayed for reference when hidden items are enabled, but they remain unavailable for selection.

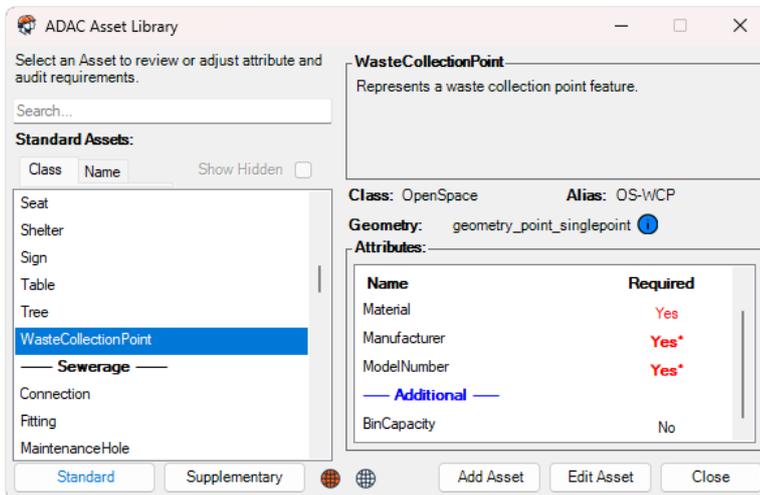
Library review workflow

When opened in library mode, the window is used to browse Standard and Supplementary asset definitions, inspect their structure and access maintenance tools. The maintenance tools themselves are described elsewhere in the manual.



Screen layout

The Asset Library window is arranged to keep browsing on the left and asset review on the right. The left panel contains the search field, sorting tabs and the asset list. The right panel presents the selected asset's details in a fixed summary layout, followed by the attributes list. Bottom buttons change depending on whether the window is being used for Create, Generate or library review.



Left panel

The left panel is the main navigation area for the Asset Library. It is used to narrow down the available assets and choose the item to review.

- **Search** - The Search field filters the asset list to help locate assets more quickly. Search can match against asset name and related definition content, making it useful when the exact asset name is not known. Clearing the Search field restores the full list.
- **Sort tabs** - The tabs allow the asset list to be viewed in different ways. **Name** sorts assets alphabetically by asset name. **Class** sorts assets by class and then by name, with divider rows inserted between class groups. Divider rows are headings only and cannot be selected.
- **Show Hidden** - In Generate mode, a **Show Hidden** checkbox may be displayed beside the tabs. When enabled, rejected assets that are normally hidden can be shown in the list for reference. These rejected assets remain disabled and cannot be generated.

Right panel

The right panel displays the definition details of the currently selected asset. This area is intended for review rather than navigation and updates as the user changes the selection on the left.

- **Asset name** - Displayed as the header at the top of the right panel.
- **Description** - Displayed in a bordered scrollable area so longer descriptions can be read without disrupting the layout.
- **Alias and Layer** - Shown together in a paired row beneath the description. Long values are truncated visually where required to keep the layout stable.
- **Class** - Displays the class or feature grouping associated with the selected asset.
- **Geometry** - Displays the stored geometry definition for the asset. Where an asset supports more than one geometry, the field displays **Multiple Geometries** rather than listing every geometry token directly in the field.



Asset list indicators

The asset list uses colour and row style to help communicate the source or status of an asset at a glance. These indicators are intended to help the user understand whether an asset is part of the base library, an additional asset or unavailable in the current context.

- **Green asset name** - Indicates an asset that was added or cloned during the current session.
- **Blue asset name** - Indicates an additional or extension asset loaded from profile data or a user override source rather than the embedded base set.
- **Grey asset name** - Indicates a rejected asset for the current Create or Generate context. In Generate mode, these may appear when hidden items are shown, but they cannot be selected.

Geometry status icon and tooltip

- A status icon is displayed beside the Geometry field. Its role depends on the mode in which the Asset Library window is being used.
- In library review mode, the icon is informational and is used to describe the geometry stored against the selected asset. If the asset supports multiple geometries, the tooltip provides a fuller explanation of the configured geometry tokens and their meaning.
- In Create or Generate workflows, the same area helps communicate whether the selected asset's geometry is suitable for the current context. This allows the user to review the asset before attempting to continue.

Attributes list

- The attributes list displays the attribute schema for the selected asset. It is intended to help the user understand what information is associated with the asset and which attributes are currently marked as required.
- Required attributes are visually highlighted so they can be identified more quickly. Nested attributes are shown using a path-style naming format so it remains clear which selectable option or parent attribute they belong to. Where the asset includes additional attributes or notes, these are separated into their own section in the list so they can be distinguished from the core attribute set.
- For Standard assets that exist in the embedded standard comparison set, the Required status may also reflect differences between the loaded profile requirement and the embedded ADAC standard expectation for the current standard version. These differences are shown using the existing text and icon behaviour in the list.

Attribute tooltips

Attribute tooltips are provided to keep the list readable while still making full detail available when needed. Tooltips only appear when an attribute row is selected and the pointer is positioned over that row. This helps reduce clutter and prevents constant tooltip popups while moving through the list.

The tooltip may include:

- The attribute name
- The attribute type
- The attribute description
- The current required status
- Any note related to standard comparison or requirement differences



Standard and Supplementary views

In library mode, the Asset Library can switch between Standard and Supplementary asset sets. This allows the user to browse the two libraries independently while keeping the same review layout on screen.

- **Standard** - Displays the primary standard asset definitions associated with the active profile and standard context.
- **Supplementary** - Displays the supplementary asset definitions that sit outside the core standard library.

Switching between these views refreshes the list and updates the available asset set shown in the left panel.

Profile-dependent behaviour

Some Asset Library capabilities depend on a profile being loaded. If no profile is loaded, the window may operate in an embedded resource mode. In this mode, the user can still browse assets and inspect their details, but profile-dependent management functions are restricted.

This behaviour ensures the library remains available for review even when editing functionality cannot be enabled.

Selection behaviour

The Asset Library is designed to support asset selection as well as review. In Create and Generate workflows, only valid assets can be confirmed. Divider rows are never selectable and rejected assets cannot be confirmed even when visible.

This means the user can still inspect unavailable assets when needed but cannot accidentally proceed with an invalid selection.

Tips

The following points can help when using the Asset Library efficiently:

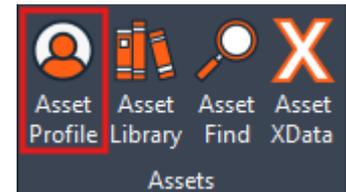
- Use **Name** view when you already know the asset name and want the fastest lookup.
- Use **Class** view when browsing by discipline, class or feature grouping.
- Use **Search** when only part of the asset name, description or related definition is known.
- Check the **Geometry** field and its tooltip before using an asset in a Create or Generate workflow.
- Review the **Attributes** list before selection if you need to confirm what fields are required.
- If an item appears greyed out, it is being shown for reference only and is not currently selectable in that workflow.



Asset Profile Window

The **Profile** window is the form opened by the '**ADACProfile**'  command. Its purpose is to open the **currently loaded profile** in a dedicated profile form so the user can review its details and work with that profile directly.

This window is separate from the Settings window profile panel. The Settings panel is mainly for selecting and managing profiles, while the Profile window opens the active profile itself in its own dialog.

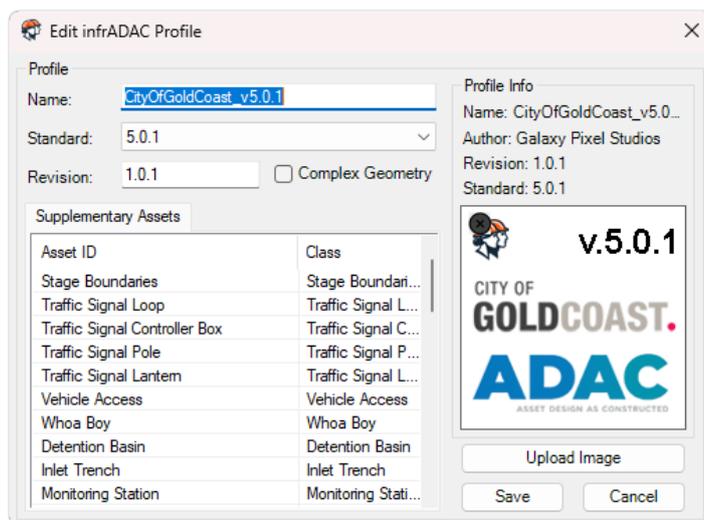


Opening the Profile window

The Profile window is opened by running the '**ADACProfile**'  command.

Before the window opens, **infrADAC** checks that licence validation and disclaimer validation have both passed. If either check fails, the command does not continue.

When the command proceeds, **infrADAC** reads the current profile recorded in settings and then checks the temporary loaded profile folder. The window relies on the currently loaded profile already being available in that working location. If the required loaded profile folder or profile information file cannot be found, the window does not open and an error message is shown.



Window purpose

The Profile window is used to display the currently loaded profile in a dedicated form. It loads the profile data already extracted into the temporary **LoadedProfile** location and presents it through the profile form controls.

This means the window works with the active profile that **infrADAC** is currently using, rather than asking the user to browse for a profile package first.

What is loaded into the window

When the Profile window opens, **infrADAC** loads the following current profile content:

Current profile data

The currently active profile name is read from settings and used to load the profile content into the form.



Profile information

The profile information is read from the **info.json** file stored in the temporary loaded profile folder. This is used to populate the profile details shown in the form.

Supplementary assets

The profile standard is read from the loaded profile information, and **infrADAC** then looks for the matching **SupplementaryAssets** file for that standard. If found, the supplementary assets are loaded and passed into the profile form so they can be shown in the supplementary assets list.

If no supplementary asset file is found, the window still opens, but supplementary asset content may not be shown.

Save and Cancel controls

At the bottom of the Profile window are the main action buttons:

- **Save**
- **Cancel**

These controls are part of the profile form and behave in line with the profile editing workflow. Because the creation and editing process is already covered elsewhere, that detailed behaviour should be referred to in the dedicated profile editing section rather than repeated in full here.

For this section it is enough to note that the window allows the currently loaded profile to be worked on directly, and those actions are completed through the Save and Cancel controls in the form.

Complex Geometry option

The Profile window also includes a **Complex Geometry** checkbox.

This setting is part of the profile definition and indicates whether the profile accepts complex geometry. When the current profile is loaded into the window, this option is populated from the profile data if that value exists.

Error handling

If **infrADAC** cannot find the currently loaded profile folder, the profile information file or other required loaded profile content, the Profile window does not open successfully and an error message is displayed.

Likewise, if an unexpected AutoCAD or general application error occurs while opening the command, the command stops and an error dialog is shown.



Settings

The Settings window is the central location for configuring how **infrADAC** behaves, where it stores working folders, how much information is written to the log, which automatic setup actions are used for ADAC New workflows, how licensing is managed and how software updates are checked. It also provides access to the current profile area on the right side of the window.



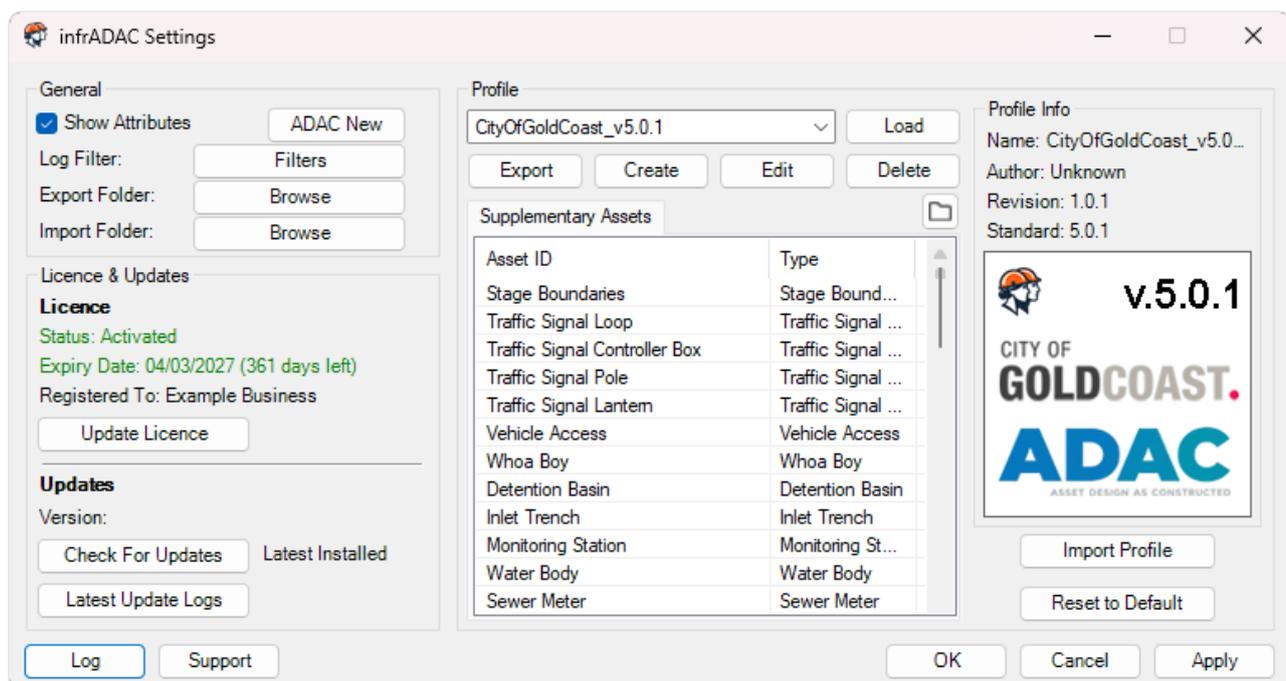
Most settings can be changed at any time and saved immediately. However, if the licence is invalid, expired or not activated, **infrADAC** restricts most non-licence controls so that only licence recovery actions remain available. This helps ensure that licence issues can still be corrected even when the rest of the settings window is locked down.

Window Layout

The Settings window is designed so that routine behaviour settings are grouped separately from licence and update functions. This makes it easier to find the setting you need without moving between unrelated controls.

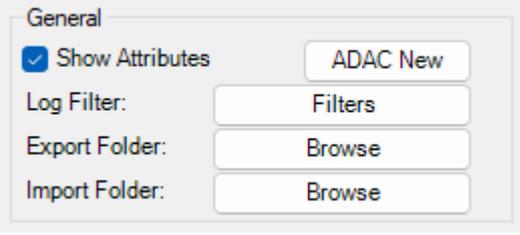
The left side of the window contains two grouped sections. The **General** section controls everyday behaviour such as whether attribute forms appear during insertion, which log levels are recorded and which default folders are used for export and import. The **Licence and Updates** section shows the current licence state and provides access to licence refresh and software update tools.

The right side of the window contains the **Profile** area. This part of the window is used to view and manage the currently available profile information and related profile actions.



General

The General group contains the most commonly used operational settings. These options affect how **infrADAC** behaves during day-to-day use and where it stores or retrieves working files.

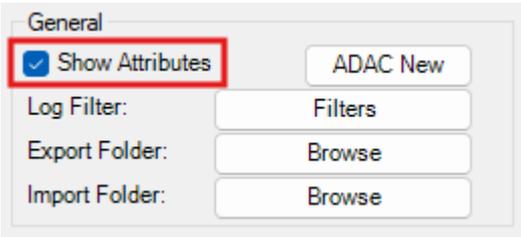


Show Attributes

The **Show Attributes** option controls whether the asset attribute form opens during asset insertion.

When enabled, **infrADAC** displays the attribute form so the user can review and enter attribute values during insertion. This is useful when data entry needs to happen immediately as assets are placed.

When disabled, assets can be inserted without opening the attribute form. This is useful for faster drafting workflows where immediate attribute entry is not required.



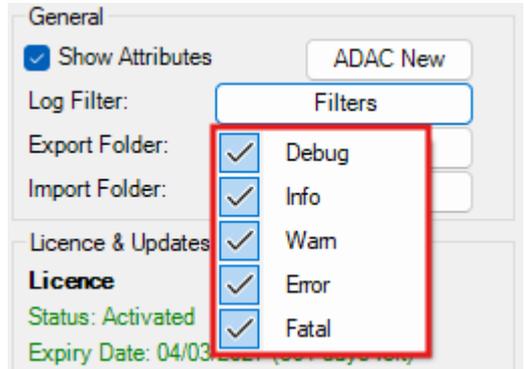
Log Filter

The **Log Filter** setting controls which message levels are written to the **infrADAC** log file. Selecting **Filters** opens a checklist menu where each log level can be enabled or disabled independently.

This allows the log to be tailored for either routine use or detailed troubleshooting. Reducing lower priority entries can keep log files smaller and easier to review, while enabling more detailed levels can help with diagnostics.

The available log levels are:

- **Debug** - Detailed diagnostic information intended mainly for troubleshooting and development investigation.
- **Info** - Normal operational messages that record routine actions and successful processes.
- **Warn** - Unexpected conditions or non-fatal issues that did not stop processing but may need attention.
- **Error** - Failures that prevented an action from completing correctly.
- **Fatal** - Critical failures that may prevent **infrADAC** from functioning correctly.

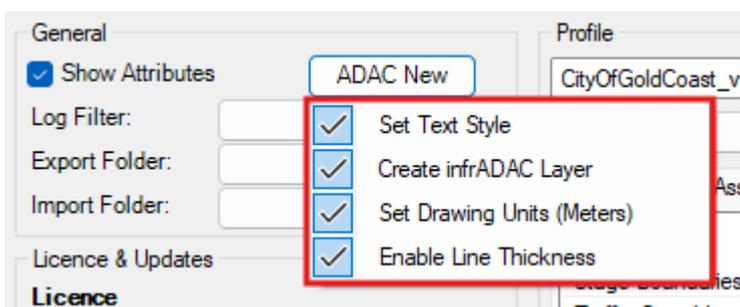



ADAC New

The **ADAC New** button opens a checklist of optional actions that can be applied when preparing a drawing for ADAC work. Each option can be enabled or disabled independently and the selected state is saved with the user settings.

These options allow **infrADAC** to apply consistent setup steps automatically when starting ADAC work. The available items are:

- **Set Text Style** - Applies the text style configuration expected for **infrADAC** and ADAC workflows.
- **Create infrADAC Layer** - Creates the standard **infrADAC** layer if it does not already exist.
- **Set Drawing Units (Meters)** - Sets drawing units to metres to align with typical ADAC requirements.
- **Enable Line Thickness** - Turns on line thickness display for improved visual review and plotting consistency.



Export Folder

The **Export Folder** setting defines the default folder used when **infrADAC** exports files. This can include exported project outputs, generated files and other saved results that rely on the configured export location.

Select **Browse** to choose a new export folder. The selected folder is stored in the user settings and becomes the default export location until it is changed again.

Import Folder

The **Import Folder** setting defines the default folder used when importing files into **infrADAC**. This can include profile packages and other supported **infrADAC** data files.

Select **Browse** to choose a new import folder. The selected folder is stored in the user settings and becomes the default import location until changed.

Folder path storage and %LOCALROOTPREFIX%

Export and import paths support the **%LOCALROOTPREFIX%** token. When a chosen folder sits under the AutoCAD/BricsCAD LOCALROOTPREFIX location, **infrADAC** can store the path using this token rather than saving the full absolute path.

This improves portability across installations and user environments. When **infrADAC** reads the settings back, the token is expanded automatically to the current BricsCAD LOCALROOTPREFIX value.



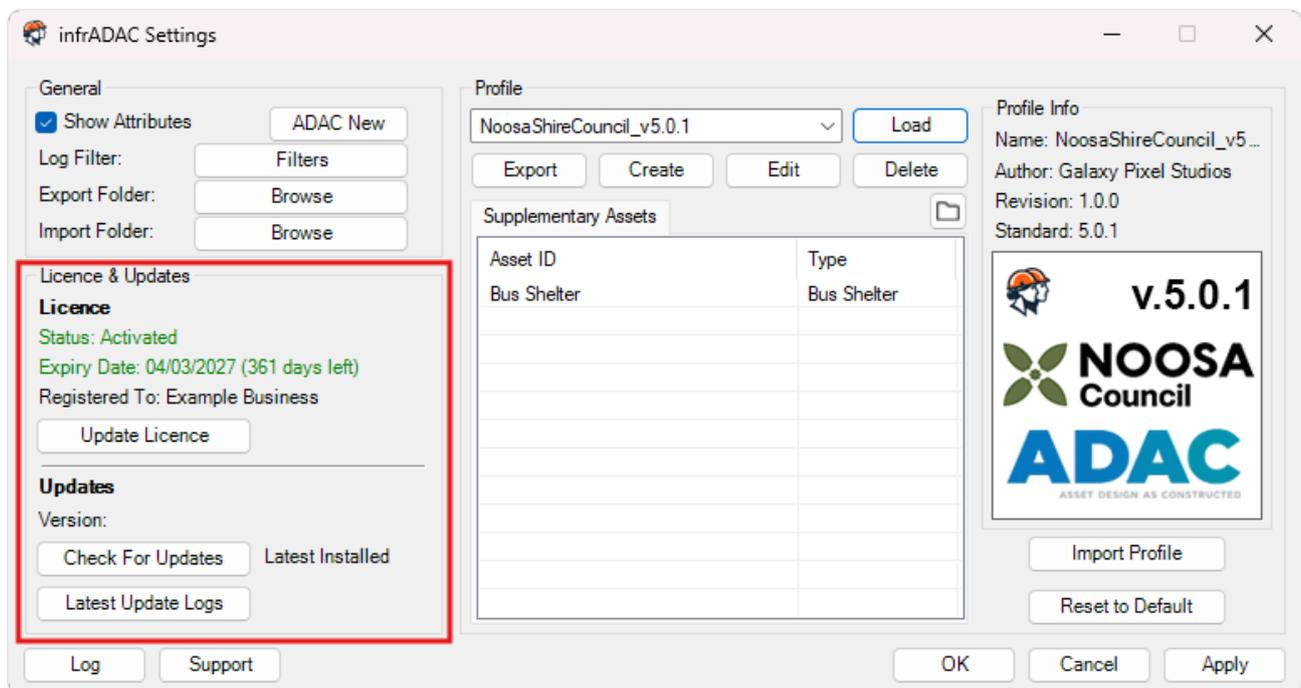
Bottom action buttons

The bottom row provides quick access to common actions related to logs, support and saving changes.

- **Log** - Opens the latest **infrADAC** log file. This is useful for troubleshooting, support requests and reviewing recent activity.
- **Support** - Opens the Support Ticket form so assistance can be requested directly from within **infrADAC**.
- **OK** - Saves pending changes and closes the Settings window.
- **Cancel** - Closes the Settings window without saving any unsaved changes made during the current session.
- **Apply** - Saves pending changes without closing the Settings window.
- **Help** - Reserved for help content.

Licence and Updates

The Licence and Updates group is used to review the current licence state, refresh the installed licence file and access software update tools.



Licence information

The licence section displays the current activation state and the key licence details available to the user. This includes:

- **Status** - Shows whether the licence is activated, invalid, expired or otherwise unavailable.
- **Expiry Date** - Shows the licence expiry date and, where relevant, the remaining number of days. Lifetime licences are shown as **Lifetime**.
- **Registered To** - Shows the business name recorded in the licence data.

The expiry display is colour-coded to help indicate licence health. A healthy licence is shown in green, licences nearing expiry move through warning colours and expired or invalid licences are shown in red.



Behaviour when the licence is invalid or expired

If the licence is invalid, expired or not activated, **infrADAC** disables most settings controls and profile controls. This prevents configuration changes in areas that require a valid licence.

In this state, the following actions remain available so the issue can still be corrected:

- **Update Licence**
- **OK**
- **Cancel**
- **Apply**

Software update controls are also disabled when the licence is not in a valid active state.

Update Licence

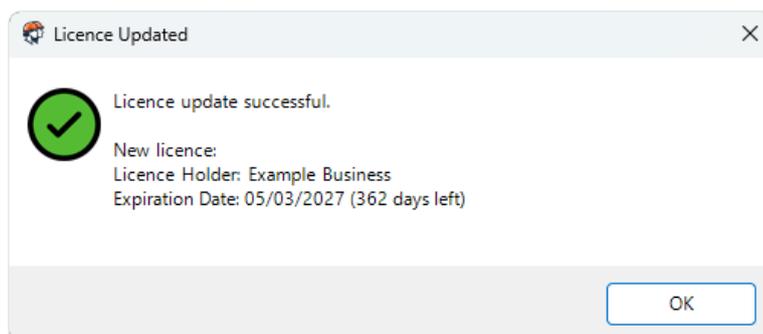
Selecting **Update Licence** prompts the user to choose how the licence should be refreshed. Two update paths are supported:

1. **Server update** - The server option downloads the latest licence file from the configured licence service, validates it and installs it if successful.
2. **Local file update** - The local file option allows the user to browse to a licence.dat file and install it manually after validation.

The local update process includes safeguards to avoid meaningless replacements. **infrADAC** blocks:

- selecting the licence file that is already installed
- selecting a licence file that is effectively identical to the installed licence
- selecting a licence file that does not provide the required updated expiry in situations where a refreshed licence is required

If a valid updated licence is installed successfully, the licence details in the Settings window are refreshed immediately.



Updates

The updates section displays the installed software version and provides access to update-related tools.

- **Check For Updates** - Opens the software update window so the current installation can be checked and updated if required.
- **Latest Update Logs** - Opens the update logs window so recent release notes and changes can be reviewed.

If profile content is updated as part of a software update, the Settings window reloads its profile UI so the refreshed profile content is available immediately.



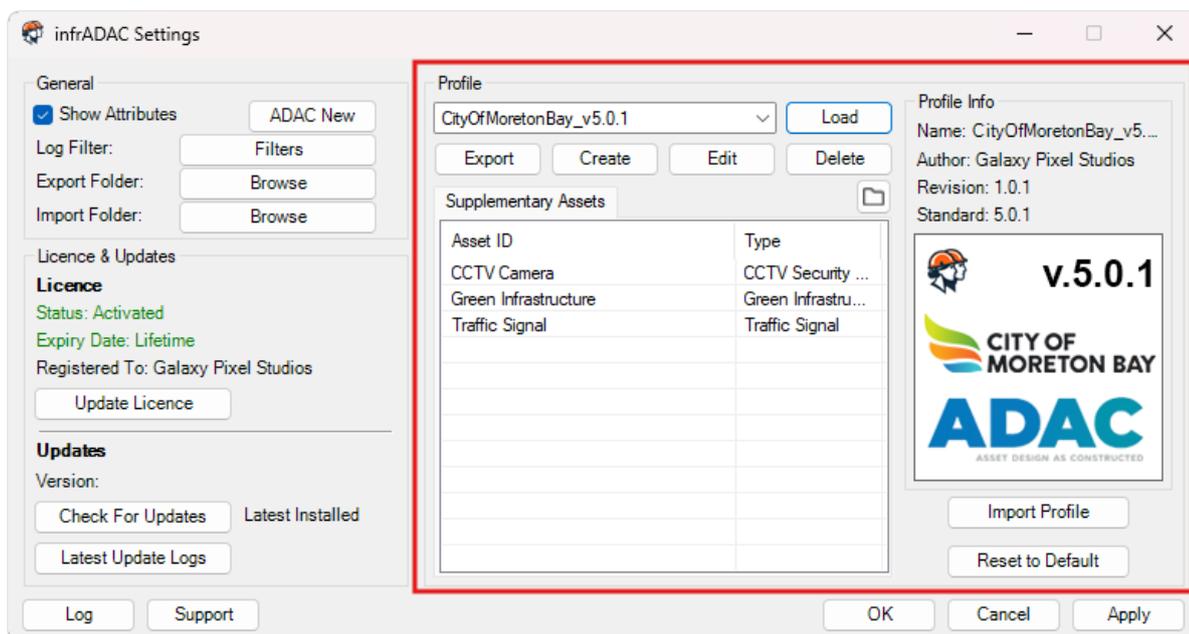
Profiles

The **Profiles** area on the right side of the **Settings** window is used to select, load, review and manage **infrADAC** profiles.

Profiles control the packaged working content used by **infrADAC**. This includes the active ADAC standard, packaged profile information, profile imagery and any supplementary asset content included with the selected profile. When a profile is loaded, **infrADAC** extracts the selected profile package into a temporary **LoadedProfile** folder, reads the profile metadata and updates the current active profile recorded in the user settings.

This area is intended to let the user do two things:

- manage available profiles
- inspect the content of the currently selected or currently loaded profile



Profile selector

At the top of the panel is the **profile selection list**. This drop down displays the profiles currently available to **infrADAC**.

The list is built from the profiles available in the configured profile location. Profiles are shown by name, and the list includes a visual divider separating normal profiles from packaged default profiles.

Important behaviour of the profile list includes the following:

- only valid profiles can be selected
- the divider line is not a selectable profile
- the currently active profile is selected automatically when the Settings window opens
- if the current profile cannot be found, the first available valid profile is selected instead

Default packaged profiles are protected differently from normal profiles. They can still be selected and loaded, but they are treated as protected items for management actions such as deletion.



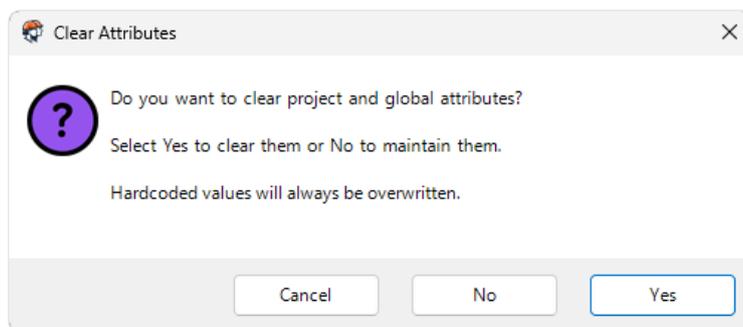
Load button

The **Load** button loads the profile currently selected in the profile list.

When a new profile is loaded, **infrADAC**:

- confirms that a profile has actually been selected
- prevents the user from loading the profile that is already active
- asks the user to confirm that the new profile should be loaded
- warns that unsaved data will be lost
- gives the user the option to clear project and global attributes before loading
- extracts the selected profile package into the temporary **LoadedProfile** folder
- reads the profile standard from the extracted profile data
- updates the current profile information saved in settings
- refreshes the profile details shown in the panel

If the user chooses to clear attributes during this process, existing project and global attributes are removed before the new profile is applied. If the user chooses not to clear them, those values are retained, although profile hardcoded values are still overwritten as required.



Profile management buttons

Below the profile selector is a row of buttons used to manage profiles.

Export button

The **Export** button exports the currently active profile to an **IDC** file chosen by the user.

This allows a profile package to be copied out for backup, transfer or reuse elsewhere. The export uses the currently loaded profile rather than only the profile selected in the list.

Create button

The **Create** button opens the profile creation workflow for making a new profile.

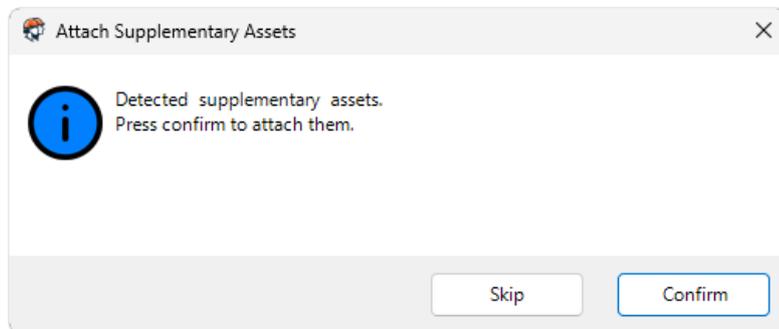
Before the creation window opens, **infrADAC** displays a small options window that allows the user to choose what should be brought into the new profile. Depending on the currently loaded profile content, the available options may include:

- **Clone Loaded Profile**
- **Merge Supplementary Assets**
- **Merge User Overrides**



If clone is selected, the new profile is based on the currently loaded profile and the other relevant options are automatically included. If supplementary assets or user override data are available, they can also be carried into the new profile when required.

This allows a new profile to be created either from a clean starting point or from an existing loaded profile with its related content.



Edit button

The **Edit** button opens the selected profile in the profile editing workflow.

If the selected profile is the currently loaded profile, the edit window uses the already loaded profile content. If a different profile is selected, **infrADAC** temporarily extracts that profile so its supplementary data and override content can be loaded into the editor.

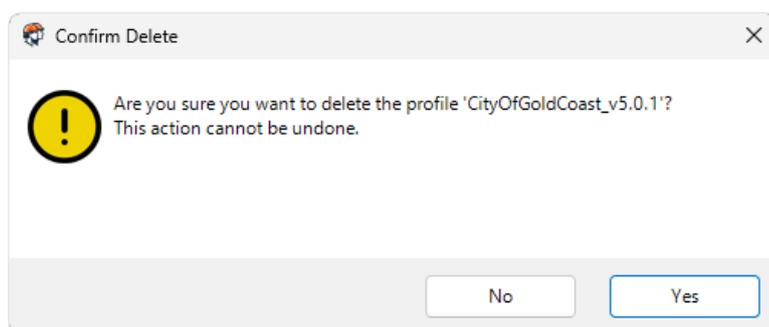
This allows the user to edit either the active profile or another available profile from the list.

Delete button

The **Delete** button deletes the selected profile.

This action is only available for profiles that are allowed to be deleted. Protected default profiles cannot be deleted. For that reason, the Delete button is automatically disabled when a protected default profile is selected.

If the user attempts to delete a normal profile, **infrADAC** asks for confirmation before deleting it. If the profile being deleted is also the currently active profile, **infrADAC** automatically reverts the system back to the **DefaultProfile** so that a valid profile remains loaded.



Import Profile button

The **Import Profile** button imports a profile package into the local profile location.

This is used to bring an external **IDC** profile file into **infrADAC** so it becomes available in the profile list. Once imported, the list of available profiles is refreshed so the newly imported profile can be selected and loaded.



Reset to Default button

The **Reset to Default** button resets the active profile back to the **DefaultProfile**.

This action is useful when the user wants to return to the base packaged profile rather than continue using a custom or imported profile. **infrADAC** asks for confirmation before resetting and warns that unsaved data will be lost. If the reset proceeds, the default profile is extracted, saved as the current active profile and reloaded into the profile panel.

If the default profile is already active, **infrADAC** reports that no reset is required.

Profile folder button

Next to the tab area is a small **folder button**. This is used to view or change the profile folder location.

When selected, the user can choose a different folder location for profile storage. If a new location is chosen, **infrADAC** asks whether existing profiles should also be moved to that location.

The user can choose to:

- move the existing profiles to the new location
- change the folder location without moving existing profiles
- cancel the change

After the folder location is updated, the profile list is refreshed to reflect the profiles available in the current profile folder.

Profile Info panel

On the right side of the profile area is the **Profile Info** panel. This panel displays summary information about the selected and loaded profile.

The panel includes the following fields:

- **Name**
- **Author**
- **Revision**
- **Standard**
- **Profile icon**

These details are read from the profile metadata after the profile package is extracted. If a profile icon is included in the package, it is shown in the image area. If no custom icon is provided, a default profile image is displayed instead.

Where text is too long to fit fully within the available label width, the displayed text is shortened and the full value is available as a tooltip when hovered over.

Supplementary Assets tab

The main tab currently shown in the profile area is the **Supplementary Assets** tab.

This tab displays supplementary asset content included with the selected profile, if that content exists. The list is presented in a two column view:

- **Asset ID**
- **Type**



Each entry represents one supplementary asset defined in the profile package. The list is automatically populated when the profile information is loaded.

If the profile does not contain supplementary assets, the tab displays a message indicating that no supplementary assets are available.

Column widths are controlled automatically so that the list remains readable and consistent.

Supplementary asset inspection

Supplementary assets in the list can be inspected more closely.

When an item in the supplementary asset list is activated, **infrADAC** reads the stored details for that asset and displays its properties in the supplementary assets area. This gives the user a way to inspect the data carried by an individual supplementary asset rather than only seeing its identifier and type in the main list.

If text in the list is too long to fit inside a visible column, hovering over that item displays the full text in a tooltip.

What happens when a profile is loaded

When a profile is loaded successfully, several parts of the system are updated together.

The selected profile package is extracted into a temporary working location and the metadata stored in the profile package is read. **infrADAC** then updates the stored active profile settings so the loaded profile becomes the current working profile. This includes the current profile name and the profile standard.

Once loaded, the profile panel is refreshed to show:

- the selected profile name
- author information
- revision information
- active standard
- profile icon
- supplementary asset data where available

This ensures the profile area reflects the actual currently loaded profile rather than only the last item clicked in the list.

Default profiles and protected behaviour

Some profiles are treated as packaged default profiles. These are displayed separately in the profile list and are protected from deletion.

Because of this, the panel behaves differently for those profiles:

- they can be selected and loaded
- they cannot be deleted
- the Delete button is disabled when one is selected
- the system can always fall back to the default profile if another profile is removed

This protection helps ensure that **infrADAC** always retains at least one valid baseline profile.



Profile updates and refresh behaviour

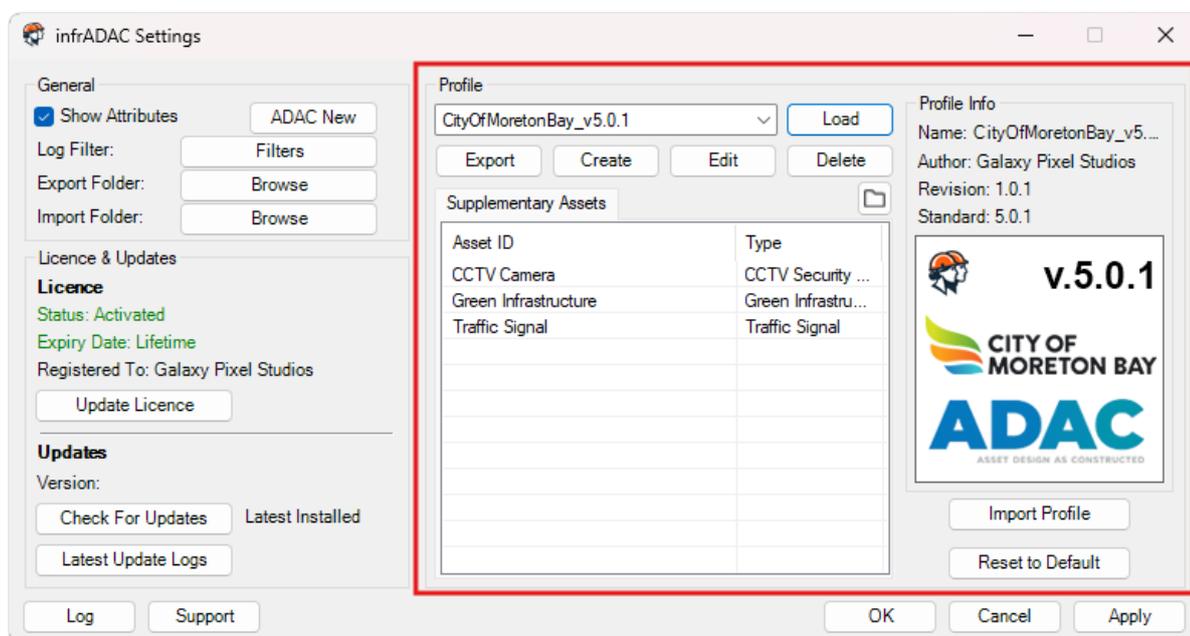
The profile panel can be refreshed after profile updates so that newly imported, updated or changed profile content is shown correctly.

This means that after profile related updates are performed, the Settings window can rebuild the profile panel and reload the settings data so the displayed profiles, metadata and packaged content match the latest saved files.

Behaviour when infrADAC is unavailable due to licensing

If **infrADAC** is not in an active licensed state, the profile area is disabled along with other non licence controls.

When this happens, the profile information display is cleared and the panel is shown in a disabled appearance. For example, profile text and supplementary asset entries are greyed out and the profile image is also visually muted. This makes it clear that profile actions are not available until the licence issue is resolved.



Saving behaviour

infrADAC stores settings in a user settings JSON file. On first use, the user settings file is created from the default settings file. When newer versions of **infrADAC** introduce additional settings, missing properties are automatically backfilled from the defaults into the user settings file without overwriting the user's existing values.

This means existing preferences are preserved while still allowing new settings introduced by updates to appear automatically.

Selecting **Apply** or **OK** writes changes to disk only when a real change is detected. If nothing has changed, **infrADAC** reports that no changes were detected and does not rewrite the settings file unnecessarily.

The following types of changes can trigger a save:

- changed checkbox settings
- changed log filter selections



- changed ADAC New options
- changed export or import folder locations
- pending profile selection changes

Notes on profile saving

Profile changes are not written immediately at every intermediate step. Instead, the Settings window can hold profile values as pending changes and then persist them through the normal save path. This keeps profile settings aligned with the rest of the Settings window behaviour.

When a profile is loaded, **infrADAC** stores the selected profile name, the selected standard and the extracted profile folder path into the user settings.

Troubleshooting

The following checks can help if the Settings window does not behave as expected:

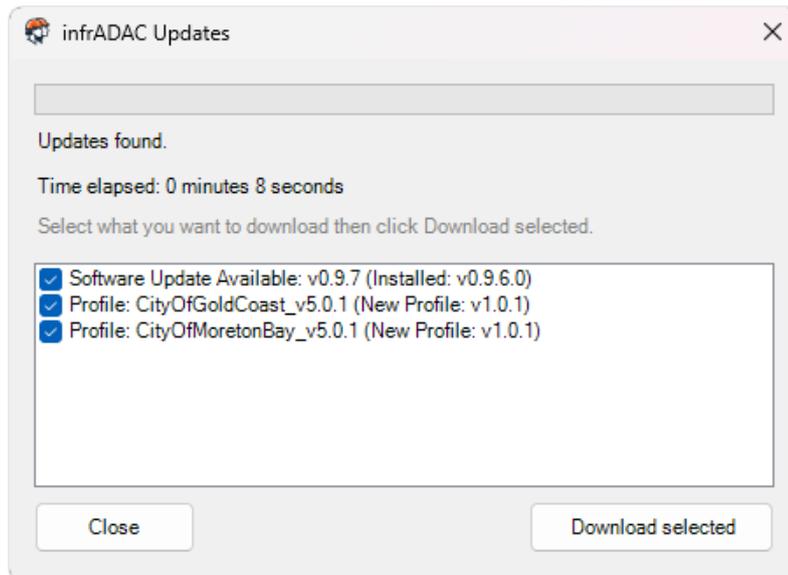
- **Controls are disabled** - Check whether the licence is valid and active. If the licence is invalid, expired or not activated, most non-licence controls are intentionally disabled.
- **Cannot use update tools** - Software update controls are disabled when the licence is not valid. Update the licence first.
- **Wrong export or import location is being used** - Review the configured Export Folder and Import Folder settings. If %LOCALROOTPREFIX% is being used, confirm that BricsCAD LOCALROOTPREFIX points to the intended root location.
- **Changes do not appear to save** - Select **Apply** or **OK** after making changes. If no real change was made, **infrADAC** will report that no changes were detected.
- **Profile information does not look current after an update** - If profiles were refreshed during a software update, the Settings window reloads the profile UI so the latest profile content can be shown.
- **Need support** - Use **Support** to open the support form and **Log** to open the latest log file for review or attachment.



Software Updates Window

The **Software Updates** window is used to check for newer **infrADAC** software releases and entitled profile package updates, then download the selected items. It is opened from the **Check For Updates** button in the Settings window.

This window is the main operational update tool. It checks online availability, compares installed versions with server versions and presents any available software or profile updates in a selectable list.



What the window does

The Software Updates window performs the following tasks:

- checks whether an internet connection is available
- queries the latest available software release
- checks entitled profile package releases
- compares installed software and installed profile revisions against server versions
- displays available updates in a checked list
- downloads selected updates
- installs profile package updates
- stages software updates for the updater process

This window is used for update action, not for browsing release history. Release notes and version summaries are handled separately in the **Update Logs** window.

Window layout

The window contains the following main elements:

- **Progress bar** - Shows activity while checking for updates or downloading selected items.
- **Status line** - Shows the current stage of the process, such as checking, downloading, verifying or completed.
- **Time elapsed** - Shows how long the current process has been running.



- **Information line** - Provides additional detail about what **infrADAC** is doing.
- **Updates checklist** - Displays available software and profile updates that can be selected for download.
- **Download selected** - Downloads the checked items.
- **Cancel** or **Close** - Cancels the current operation or closes the window once processing is complete.

Checking for updates

When the window opens, **infrADAC** checks for internet access before attempting update queries. If no connection is available, the process stops and a message is shown to the user.

If internet access is available, **infrADAC** checks for:

- a newer software release than the currently installed plugin version
- entitled profile packages with newer revisions than the installed profile revisions
- profiles that are entitled but not currently installed

If no updates are available, the window reports this and no download action is required.

Update list entries

The update list can contain both software and profile package items. Each entry describes the installed version and the server version so the user can see what is changing before downloading.

Typical list entries include:

- **Software Update Available: vX.X.X.X (Installed: vX.X.X.X)**
- **Profile: ProfileName (Update Available: vX.X.X.X)**
- **Profile: ProfileName (New Profile: vX.X.X.X)**
- **Profile: ProfileName (Rollback Update: vX.X.X.X)**

This allows the update list to represent both forward updates and cases where the server-provided package differs from the installed revision in another direction.

Download selected

After the user reviews the list, **Download selected** downloads all checked items. Software and profile updates are handled slightly differently.

Software download behaviour

Software releases are downloaded to the updater staging location and verified using SHA-256 before being accepted. Once downloaded successfully, **infrADAC** can prompt the user to launch the updater immediately.

If the updater is not launched immediately, the download can remain marked as a pending software install and the user may be prompted to run the updater the next time the Software Updates window opens.

Profile update behaviour

Profile updates are downloaded, verified and then installed into the profile storage location. Where existing user customisations are detected, **infrADAC** gives the user a choice about how those changes should be handled.



If user customisations are found in an installed profile, the user is prompted to choose one of the following:

- **Yes** - Merge user changes into the updated profile where possible.
- **No** - Replace the existing profile with the updated one and discard user changes.
- **Cancel** - Abort the profile update operation.

This allows the user to control whether customised user content is preserved or replaced during profile package installation.

Profile merge behaviour

When a profile update is installed and the existing profile contains a User folder, **infrADAC** attempts to preserve user customisations if merge is chosen. This includes three-way merge handling for relevant JSON content and preservation logic for supported user files such as blocks and line types.

The goal of this process is to carry forward user changes where possible while still applying the updated shipped profile content.

This behaviour should be described at a user level only. Internal merge rules do not need to be reproduced in full in the user guide unless specifically required.

Pending software installation

If a software update has already been downloaded but the updater has not yet been launched, the Software Updates window recognises that state when it opens. In that case, the window does not immediately run another full update check. Instead, it prompts the user to launch the updater and complete the pending installation.

If the staged update file is missing, **infrADAC** clears the pending state and asks the user to download the update again.

Cancel and Close behaviour

While an update check or download is in progress, the window can be cancelled. Once processing finishes, the button changes to a close-style behaviour so the window can be dismissed normally.

Relationship to the Settings window

This window is opened from **Check For Updates** in the Settings window. It is the detailed action window for update checking and downloading, while the Settings window only provides the entry point and a summary of the currently installed version.

Update Logs Window

The **Update Logs** window is used to browse release information for **infrADAC** software and, optionally, entitled profile packages. It is opened from the **Latest Update Logs** button in the Settings window.

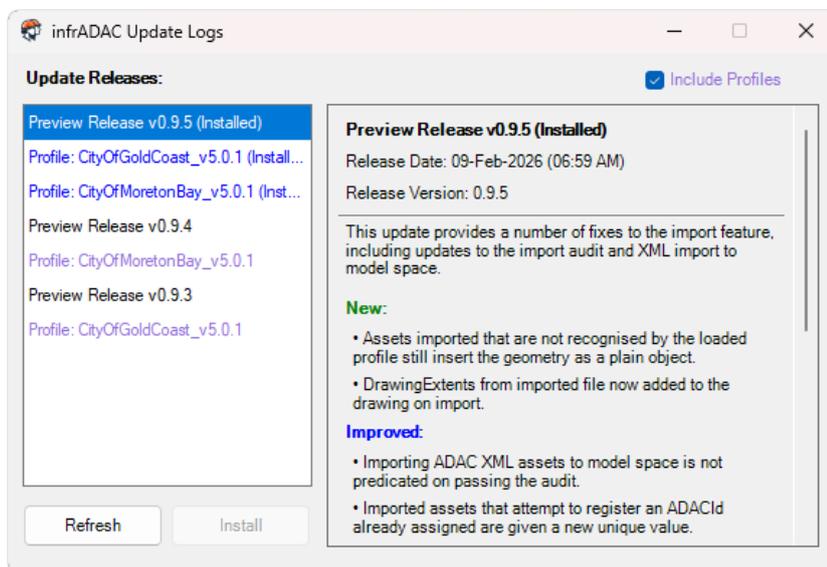


Unlike the Software Updates window, this window is primarily for reviewing release history and release notes. It allows the user to inspect what changed in each release and, where permitted, install a selected release directly from the list.

The Update Logs window is intended for release review and version comparison. It allows the user to:

- browse available software releases
- include profile package releases in the list
- inspect release descriptions
- review grouped change lists for new, improved and fixed items
- identify installed items
- install a selected eligible release directly from the window

This makes the window useful when the user wants more context about a release before deciding whether to install it.



Window layout

The window is split into two main areas.

Left side release list

The left panel contains:

- a release list
- a **Refresh** button
- an **Install** button

Each list entry represents either a software release or a profile package release. Installed items are shown differently from non-installed items and package entries are colour-coded separately from software entries.

Right side release details

The right panel shows details for the currently selected release. This includes:

- release title
- release date
- release version



- brief description
- grouped change lists under **New**, **Improved** and **Fixed**

If a section has no entries, the window displays **N/A** for that section instead of leaving it blank.

Include Profiles

A checkbox labelled **Include Profiles** controls whether entitled profile package releases are included in the release list alongside software releases.



When this option is turned on, the window queries both software release logs and profile package release logs. This gives a combined view of available release history.

Release list appearance

The release list uses visual cues to help distinguish item types and states.

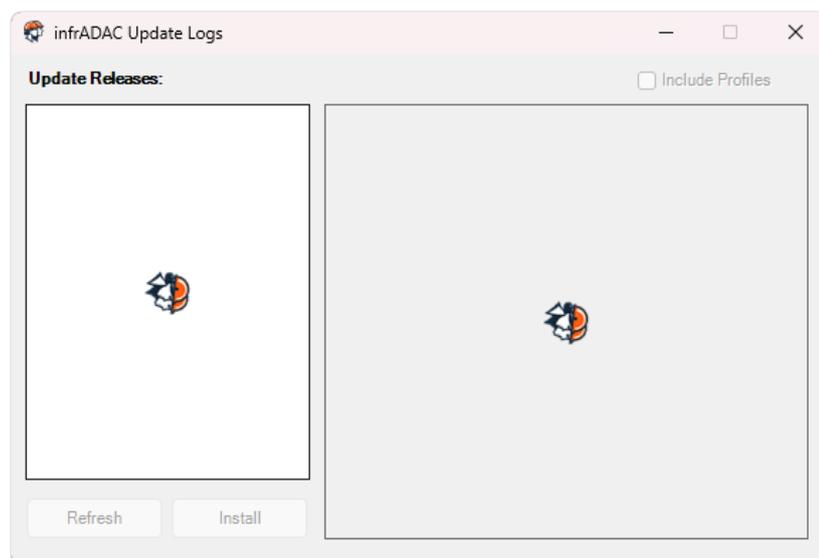
- **Installed releases** are shown as installed in the list text
- **Profile package entries** use a separate colour
- **Preview software releases** appear differently from stable releases
- the currently selected item shows its full details in the right panel

If a list item text is too long to fit, the full text is shown through a tooltip.

Refresh

Selecting **Refresh** reloads the release information from the server. This refreshes the list and the right-hand detail panel using the latest available release log data.

During refresh, loading indicators are shown over the list and detail areas so the user can see that the window is actively retrieving data.



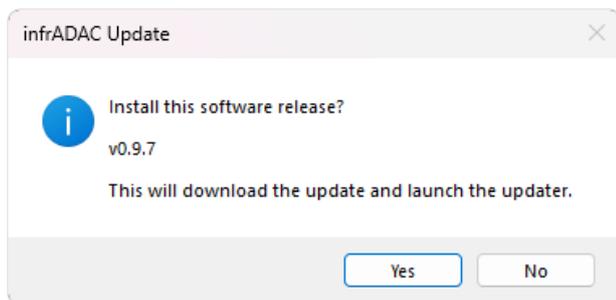
Install

The **Install** button allows the selected release to be downloaded and installed, subject to eligibility rules.

The button is enabled only when the selected release:

- is not already installed
- has a valid release identifier
- has a valid download URL
- is either a package release, the latest stable software release or a preview software release that the window allows to be installed

When selected, the user is asked to confirm installation before the download begins.



Installing from the Update Logs window

If the selected item is a profile package release, the package is downloaded and launched through the update process for package installation. If the selected item is a software release, the selected release is downloaded, verified and then passed to the updater process.

After a successful software download, the Update Logs window closes and the updater is launched.

This makes the Update Logs window both a release review tool and an alternate install entry point for eligible releases.

Release details

When a release is selected, the right panel updates to show structured release information. This typically includes:

- **Title** - The release name or profile name
- **Release Date** - Shown in local time
- **Release Version** - The release version, with preview notation where applicable and installed version context for profile packages
- **Brief Description** - A short summary of the release
- **New** - A bullet list of new additions introduced in the release
- **Improved** - A bullet list of improvements or refinements
- **Fixed** - A bullet list of resolved issues

Relationship to the Settings window

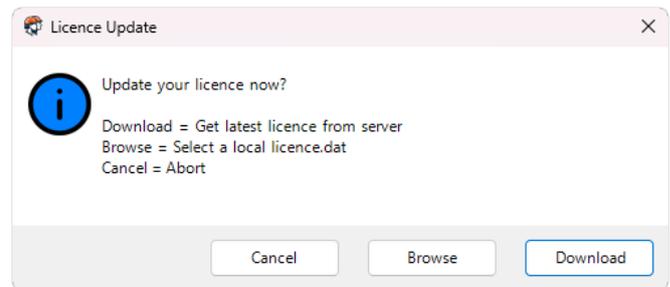
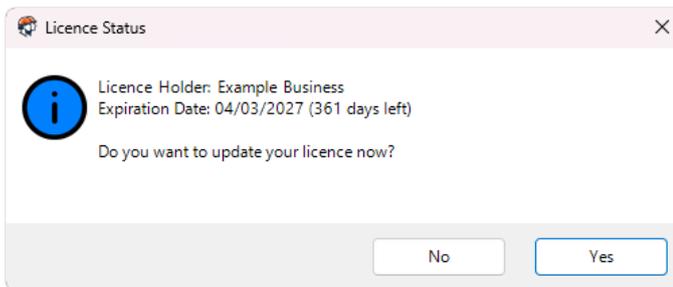
This window is opened from **Latest Update Logs** in the Settings window. It should be documented as the detailed release-history and release-details window, while the Settings window should only describe it briefly and direct the reader here for the full workflow.



Select Licence File Window

The **Select Licence File** window is a simple file selection dialog used to choose a local licence.dat file for installation. Its purpose is to prevent accidental confirmation without a valid file being selected and to provide a clear, minimal workflow for local licence updates.

This window is intended only for selecting the file path. Validation and installation are performed by the wider licence update process.



Window layout

The window is a small fixed dialog containing:

- a read-only file path field
- **Browse** button
- **OK** button
- **Cancel** button

The selected licence path is shown in the text box after a file has been chosen.

Browse

Selecting **Browse...** opens a file browser filtered to licence files with the .dat extension. This helps the user select the correct file type and reduces the risk of choosing an unsupported file.

Once a file is selected, its full path is displayed in the read-only path field.

OK

Selecting **OK** confirms the chosen file and closes the window, provided a licence file has been selected.

If the user attempts to confirm without first selecting a file, the window prevents closing and displays an error message asking the user to choose a licence file before continuing.

Cancel

Selecting **Cancel** closes the window without selecting a file and without making any changes.

Important: This window only handles file selection. It does not itself install or validate the licence. Those actions are handled by the surrounding licence update workflow.

Also note that in the current Settings workflow provided in your code, the local licence update path uses a standard file browser directly. This means this form can be documented as the dedicated local licence selection window where used, but it should not be described as always appearing unless you wire it into the active update flow.



Support Ticket Form

The **Support Ticket** form is used to send support requests, report issues and provide feedback directly from **infrADAC**. It is opened from the **Support** button in the Settings window and provides a structured way to submit the information needed for the support team to investigate and respond.



The form is intended for normal support enquiries, bug reports and general feedback. It also gives the user the option to include the latest **infrADAC** log content to help with troubleshooting.

At the top of the window, **infrADAC** displays a short guidance message explaining that the form can be used for questions, feedback or technical support. It also advises that urgent matters can be sent directly to **support@galaxypixelstudios.au**.

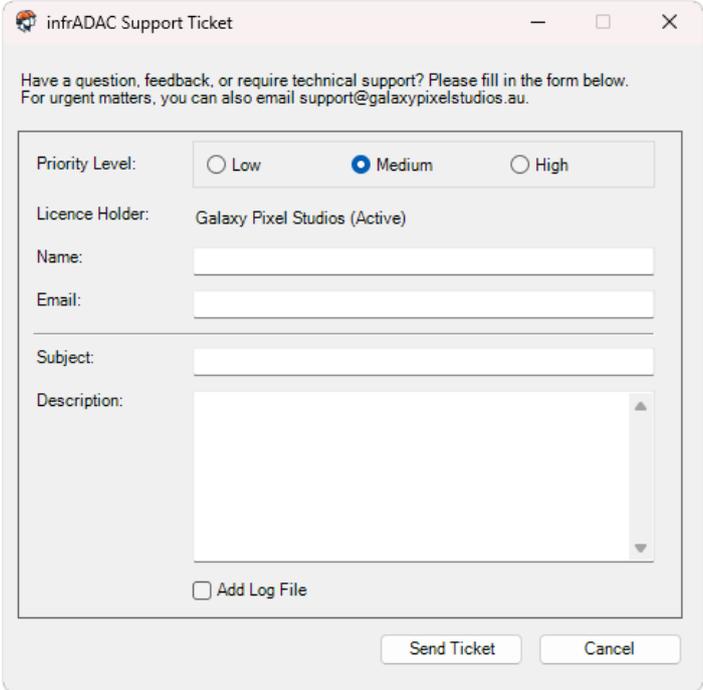
Window layout

The Support Ticket form is presented as a fixed dialog with a simple top-to-bottom layout. The main form area contains the ticket details, while the action buttons are placed below the form.

The main sections are:

- introductory message at the top
- priority selection
- licence holder information
- user contact details
- subject and description fields
- Add Log File option
- Send Ticket and Cancel buttons

This layout is designed to collect the minimum information needed to raise a support request while remaining quick to complete.



The screenshot shows the 'infrADAC Support Ticket' dialog window. At the top, there is a message: 'Have a question, feedback, or require technical support? Please fill in the form below. For urgent matters, you can also email support@galaxypixelstudios.au.' Below this is a form with the following fields: 'Priority Level' with radio buttons for 'Low', 'Medium' (selected), and 'High'; 'Licence Holder' with the text 'Galaxy Pixel Studios (Active)'; 'Name:' with a text input field; 'Email:' with a text input field; 'Subject:' with a text input field; and 'Description:' with a large text area. At the bottom left of the form area is a checkbox labeled 'Add Log File'. At the bottom right of the dialog are two buttons: 'Send Ticket' and 'Cancel'.

Priority Level

The **Priority Level** section allows the user to indicate how urgent the issue is. The available options are:

- **Low**
- **Medium**
- **High**

Medium is selected by default when the form opens.

This setting helps classify the request when it is submitted. It does not change the validation rules of the form, but it does become part of the ticket data sent to the support service.



Licence Holder

The **Licence Holder** field is populated automatically from the currently loaded licence information. This allows the support request to show which business or licence owner the request is associated with.

The displayed value includes:

- the licence business name
- the current licence state in brackets

Typical examples include:

- Business Name (Active)
- Business Name (Expired)
- Unknown (Expired) if licence details are unavailable

This field is read only and is included in the support request automatically.

Contact details

The form collects the basic contact details needed for the support team to identify the requester and respond appropriately.

The required contact fields are:

- **Name** - Enter the user's full name.
- **Email** - Enter the reply email address for the support request.

Both fields must be completed before the ticket can be sent.

Subject

The **Subject** field is used to enter a short summary of the request. This should briefly describe the issue, question or feedback topic.

Examples of suitable subject lines include:

- Asset generation error
- Profile import issue
- Licence update problem
- Feedback on Settings window

The subject field is required.

Description

The **Description** field is the main body of the ticket. It is a multi-line text box intended for a full explanation of the issue or request.

This field should be used to provide useful detail such as:

- what the user was trying to do
- what happened instead
- any error messages shown
- whether the issue is repeatable
- any relevant drawing, profile or licence context



The description field is required. Because it is the primary support content, users should be encouraged to provide enough detail for the issue to be understood without additional follow-up where possible.

Add Log File

The **Add Log File** option allows the user to include the latest **infrADAC** log content with the support ticket. This can help diagnose errors, unexpected behaviour and workflow failures.

When this option is enabled, **infrADAC** attempts to locate the most recent log file and include its contents as part of the submitted ticket. If the log content is too large, **infrADAC** trims it so it remains within the supported size limit while preserving the most recent log entries.

If the latest log file cannot be found, **infrADAC** warns the user and continues sending the ticket without the log content.

It is important to describe this feature accurately in the documentation:

- it adds the latest log content to the ticket
- it does not attach a separate file in the form UI
- if no log is available, the ticket can still be sent

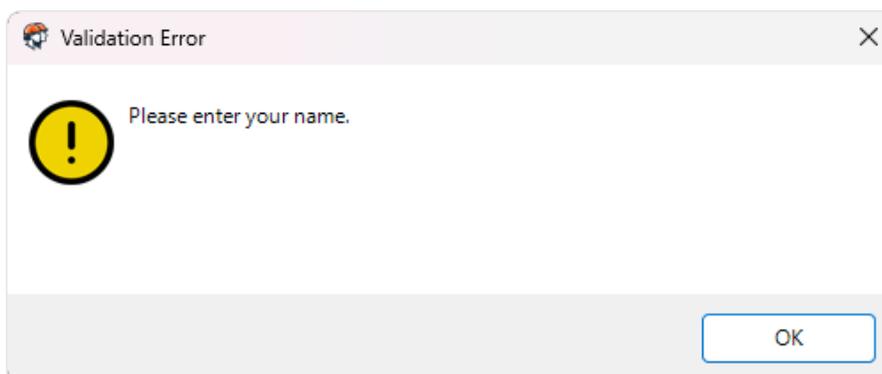
Validation before sending

Before **infrADAC** sends the ticket, the form validates the required inputs. The user cannot submit the request if any required field is missing or invalid.

The following checks are performed:

- **Name** must not be empty
- **Email** must not be empty
- **Email** must be in a valid email format
- **Subject** must not be empty
- **Description** must not be empty

If validation fails, **infrADAC** displays a warning message explaining what needs to be corrected and places focus back on the relevant field.



Sending the ticket

Selecting **Send Ticket** submits the support request to the configured support service. While the request is being sent, the **Send Ticket** button is temporarily disabled so the user cannot submit the same ticket multiple times by mistake.

The submitted ticket includes:

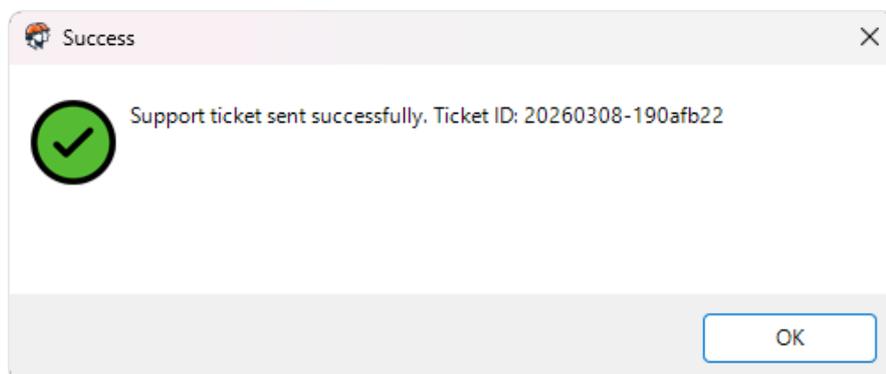
- priority level
- licence holder information
- user name
- user email
- subject
- description
- support type identifying the request as coming from the AutoCAD/BricsCAD plugin
- optional log content if selected and available

The licence holder information is included automatically as part of the submitted description so it is visible in the support record even though the field itself is read-only in the form.

Successful submission

If the support request is accepted successfully, **infrADAC** displays a confirmation message and shows the returned **Ticket ID**. The form then closes.

The Ticket ID can be used as a reference for future follow-up with support.



Failed submission

If the request cannot be sent, **infrADAC** displays an error message. This can occur for reasons such as:

- connection failure
- endpoint rejection
- invalid server response
- ticket content rejected by the support service

If the request is rejected by the support service, **infrADAC** attempts to show the returned reason so the user understands why the submission failed.

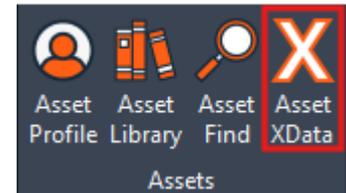
Cancel

Selecting **Cancel** closes the form without sending a support request and without saving any of the current input.



Asset XData Viewer X

The **Asset XData Viewer** is a read-only inspection window used to review the XData attached to selected entities in the drawing. It is intended for checking what **infrADAC** has stored against an asset, confirming identifier values and reviewing raw attribute data for support, auditing and troubleshooting purposes.



When the window opens, it first attempts to read the current implied selection in the drawing. If one or more entities are already selected, the viewer loads them automatically and displays their XData. If nothing is selected, the window opens with no asset loaded until the user chooses **Select Asset**.

The form does not edit any values. It is strictly a viewer for inspecting the XData currently attached to the selected entity or entities.

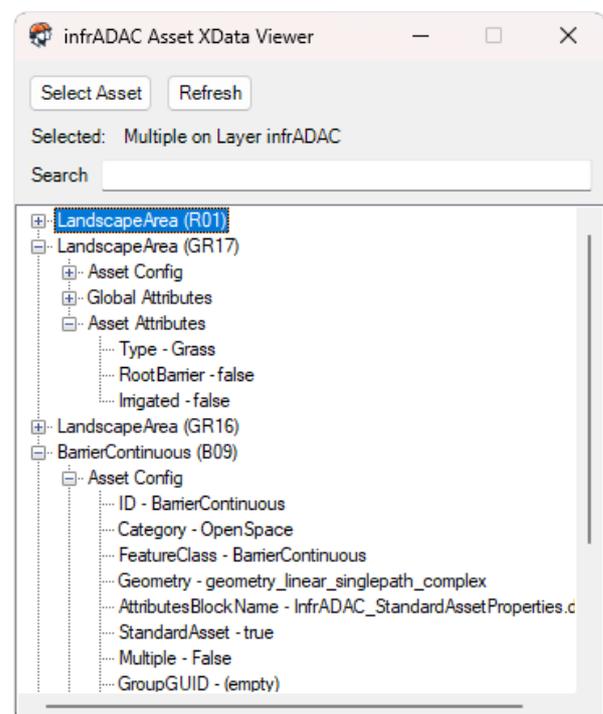
Window layout

The window is divided into two main areas.

The top section contains the selection and filtering controls:

- **Select Asset** - Prompts the user to select one or more drawing entities to inspect.
- **Refresh** - Reloads the XData for the currently loaded selection.
- **Selected** - Shows a short summary of the currently loaded asset or assets.
- **Search** - Filters the displayed tree contents as the user types.

The main lower section contains a **tree view** that displays the XData grouped by registered application name and then by individual key and value entries.



Selecting assets

The viewer can load either a single asset or multiple assets.

If the user already has entities selected in the drawing before opening the window, those entities are loaded automatically. This allows the viewer to be used immediately after a drawing selection has been made.

If the user selects **Select Asset**, the window temporarily hides while BricsCAD prompts for entity selection. Once the user completes the selection, the window reopens and reloads the XData for the chosen entities.

This makes the form suitable for:

- inspecting one asset in detail
- comparing several selected assets
- reloading the same assets after changes elsewhere in the drawing



Selected summary

The **Selected** line at the top of the form gives a quick summary of the loaded asset or assets.

For a single asset, the summary shows:

- the asset ID value
- the ADACId value
- the layer name

The summary is displayed in the format:

- AssetID (ADACId/ObjectId) on Layer LayerName

If either the Asset ID or ADACId/ObjectId value is blank, the form displays **(empty)** so the missing value is obvious.

For multiple assets, the summary changes to a broader description:

- Multiple on Layer LayerName when all selected assets are on the same layer
- Multiple on Layer multiple when the selected assets are on different layers

How XData is displayed

The tree view organises the XData into logical groups based on the registered application name, also known as the RegApp name.

For recognised **infrADAC** related RegApps, the viewer shows a friendlier display title:

- **AssetID** is shown as **Asset Config**
- **globalassetattributes** is shown as **Global Attributes**
- **InfrADAC** is shown as **Asset Attributes**

Any other registered application names are shown using their original RegApp name.

Within each RegApp section, the form displays the stored entries as individual lines in the format:

- Key - Value

If a key or value is blank, the form shows **(empty)** rather than hiding it. This makes it easier to identify incomplete or missing data.

For a single selected asset, the tree starts directly with the RegApp sections. For multiple selected assets, the tree first groups the data under an asset header for each loaded entity.

Asset identification values

The viewer extracts two important identifying values from the loaded XData so the asset can be recognised more easily.

- **Asset ID**

Taken from the **AssetID** registered application using the **ID** field.

- **ADACId/ObjectId**

Taken from the first available recognised asset attribute source. The viewer checks the common **infrADAC** related RegApps and uses the first matching ADACId/ObjectId value it finds.



This behaviour makes the window useful even where the identifier data may be stored under different but supported XData sources.

Search and filtering

The **Search** field filters the displayed tree as the user types. The search is applied live and updates the tree immediately.

The filter can match against:

- the asset-level header text in multi-selection mode
- the displayed RegApp section name
- the displayed key and value lines

This allows the user to quickly locate a specific identifier, attribute name or stored value without manually expanding and reviewing every section.

When a search term is active, the tree expands automatically so matching results are immediately visible.

If there are no matches, the tree displays **No matching results**.

Refresh

The **Refresh** button reloads the XData for the currently loaded entities. It does not change the selection. It simply rereads the XData from the current entities and rebuilds the tree.

This is useful when:

- the asset XData has been updated since the window was first opened
- the user wants to confirm the latest stored values
- the drawing has changed and the viewer needs to be synchronised again

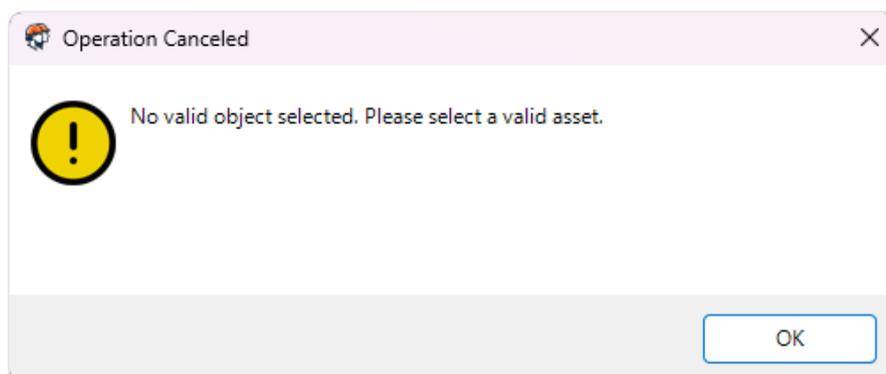
No XData found behaviour

If the selected entity or entities contain no readable XData entries, the form still loads the selection but warns the user.

The message changes depending on the number of selected assets:

- for one selected asset, the message states that the selected asset has no XData attached
- for multiple selected assets, the message states that the selected assets have no XData attached

The tree view also reflects this by displaying a simple informational line instead of empty RegApp groups.



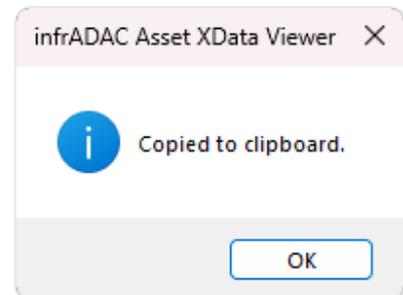
Copying values

The tree supports quick copying of displayed XData entries. If the user double-clicks a leaf entry in the tree, that full displayed line is copied to the clipboard.

This is useful when:

- recording information for support
- comparing values outside the drawing
- copying identifiers or attribute values into notes or tickets

After the copy action, the form confirms that the value was copied successfully.



Typical uses

The Asset XData Viewer is useful for:

- checking exactly what **infrADAC** has stored against an asset
- confirming that Asset ID and ADACId/ObjectId values are present
- reviewing asset configuration or attribute XData
- comparing XData across multiple selected assets
- gathering technical information for support and troubleshooting

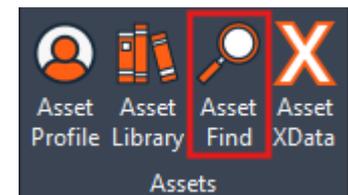
Notes and tips

- This window is a viewer only. It does not edit XData.
- If the window opens with no asset loaded, no implied selection was available at the time it opened.
- Empty values are shown as **(empty)** rather than being hidden.
- The **Selected** summary is the quickest way to confirm the loaded asset ID, ADACId/ObjectId and layer.
- If the asset contains no readable XData, the window will warn the user and show a no data message in the tree.



Find Asset Tool

The **Find Asset Tool** is used to scan the current drawing for **infrADAC** assets and present them in a searchable tree. It helps users locate assets that already exist in model space and quickly navigate to them without manually searching through the drawing.



The tool only includes entities that contain a recognised **ADACId/ObjectId** value in their XData. This keeps the results focused on identifiable **infrADAC** assets rather than showing every entity in the drawing.

The form is especially useful in larger or more complex drawings where locating a specific asset visually would be slow or impractical.

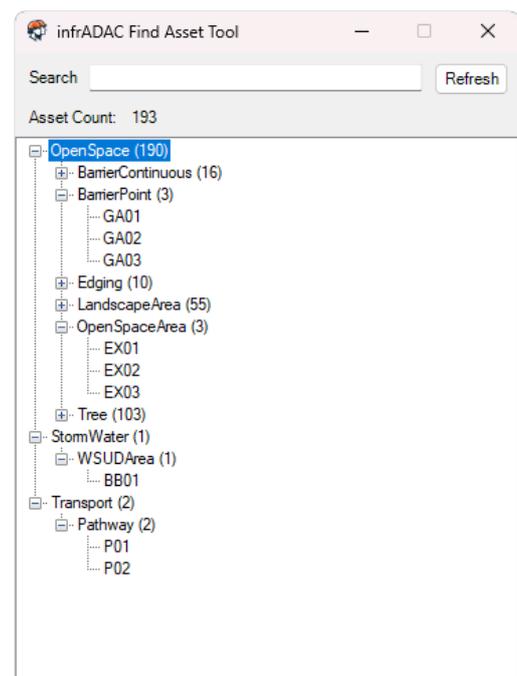
Window layout

The window contains a compact top bar and a main results area.

The top bar includes:

- **Search** - Filters the asset list as the user types.
- **Refresh** - Rescans model space and rebuilds the asset list.
- **Asset Count** - Shows the total number of qualifying assets found in the drawing.

The main lower area contains a **tree view** that groups the found assets in a structured hierarchy.



What the tool scans

When the form opens, and whenever **Refresh** is selected, the tool scans **model space** and collects all entities that contain a recognised **ADACId/ObjectId** value.

For each qualifying asset, the tool records:

- class name
- asset type
- ADACId/ObjectId
- entity handle
- layer

These values are used to group, search and locate the asset, even though only the ADACId/ObjectId is shown at the lowest visible level in the tree.

How results are grouped

The results tree is organised into three levels:

- **Class** - Top-level grouping
- **Asset Type** - Second-level grouping within each class
- **ADACId/ObjectId** - Leaf node for each individual asset



Each class node displays the number of assets in that class. Each asset type node also displays its own count. At the leaf level, only the **ADACId/ObjectId** is displayed so the list remains clean and easy to scan.

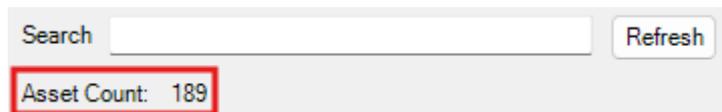
Asset Count

The **Asset Count** value at the top of the window shows the total number of qualifying assets found during the most recent scan of model space.

This count reflects the full scanned result set and is not reduced by the current search filter. In practice, this means:

- **Refresh** updates the total asset count
- **Search** changes the visible results in the tree, but the total asset count remains the count of all scanned assets

This distinction is useful when the user wants to know both how many assets exist in the drawing overall and how many of them match a current search.



Searching and filtering

The **Search** field filters the results live as the user types. Although the leaf nodes only display ADACId/ObjectId values, the search is broader than the visible text.

The filter can match against:

- class name
- asset type
- ADACId/ObjectId
- Layer
- entity handle

This allows the user to search by a value that may not be shown directly in the tree. For example, a user can search for a known layer name or entity handle and still locate the matching asset.

When a search term is active, the tree expands automatically so matching results are visible immediately.

Refresh

The **Refresh** button rescans model space and rebuilds the asset list from the current drawing contents.

This should be used when:

- assets have been created or removed
- asset identifiers have changed
- the drawing has been updated since the form was opened
- the user wants to confirm that the list reflects the latest drawing state

Before rebuilding the list, the tool also stops any temporary flashing highlight that may still be active from a previous locate action.



Selecting an asset

A single left-click on an asset leaf node selects that asset in the drawing. The tool updates the implied selection so the chosen entity becomes the current selected object in BricsCAD.

This is useful when the user wants to:

- identify the selected asset in the drawing
- inspect its properties
- prepare it for another command or review step

A single click does not zoom. It only selects the entity in the drawing.

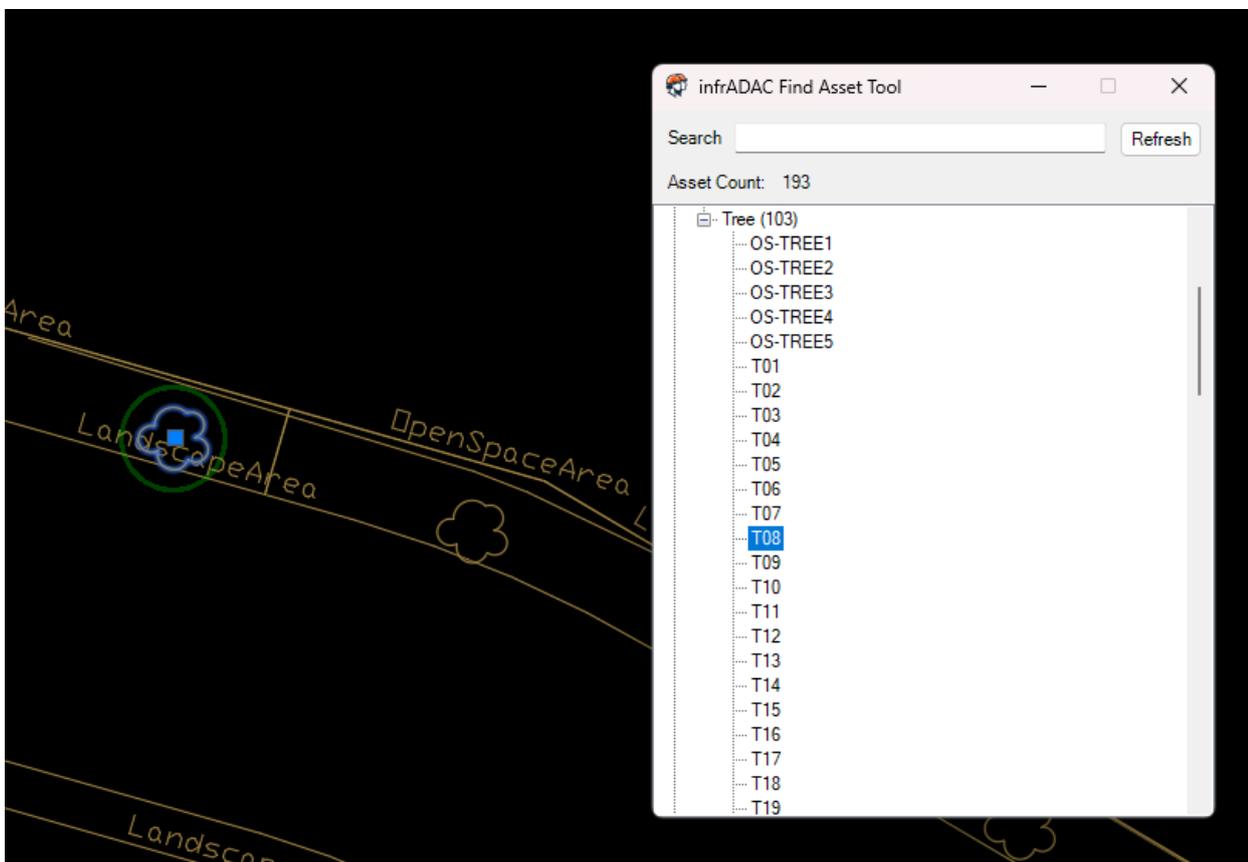
Double-click locate behaviour

Double-clicking an asset leaf node performs a full locate action. The tool will:

- zoom to the asset
- select the asset
- flash the asset in green using transient graphics

The temporary flash highlight remains active for about four seconds, making the located entity much easier to identify in the drawing.

Where the entity has valid geometric extents, the form zooms to those extents with generous padding so the asset can be seen clearly in context. If the entity does not provide usable extents, the tool falls back to a point-based zoom where possible.



How class and asset type are determined

The Find Asset Tool reads the classification data from the **AssetID** XData.

It uses:

- **ID** as the asset type
- **Category** as the preferred class name
- **FeatureClass** as a fallback class name if Category is not present

This allows the tool to group the scanned assets in a way that reflects their **infrADAC** identification data.

How ADACId/ObjectId is detected

To determine whether an entity should be included in the results, the tool checks several recognised XData sources for an **ADACId/ObjectId** value. It uses the first valid match found from the supported **infrADAC** related registered applications.

This makes the tool tolerant of supported variations in where the ADACId/ObjectId has been stored, while still ensuring that only identifiable assets are listed.

Typical uses

The Find Asset Tool is useful for:

- locating a known asset by ADACId/ObjectId
- browsing all **infrADAC** assets in model space by class and type
- quickly navigating to an asset in a large drawing
- checking how many identified assets exist in the drawing
- confirming whether a specific asset is present

Notes and tips

- The tool scans **model space only**.
- Only entities with a recognised **ADACId/ObjectId** are included.
- The leaf nodes display only the ADACId/ObjectId, but the search also works against class, type, layer and handle.
- A single click selects the asset in the drawing.
- A double-click zooms to the asset, selects it and flashes it temporarily.
- If the list does not reflect recent drawing changes, use **Refresh** to rescan the drawing.



Help Window ?

The **Help Window** provides access to the infrADAC user manual from within the program. It allows the manual to be viewed without leaving the application, making it easier to check instructions while working.

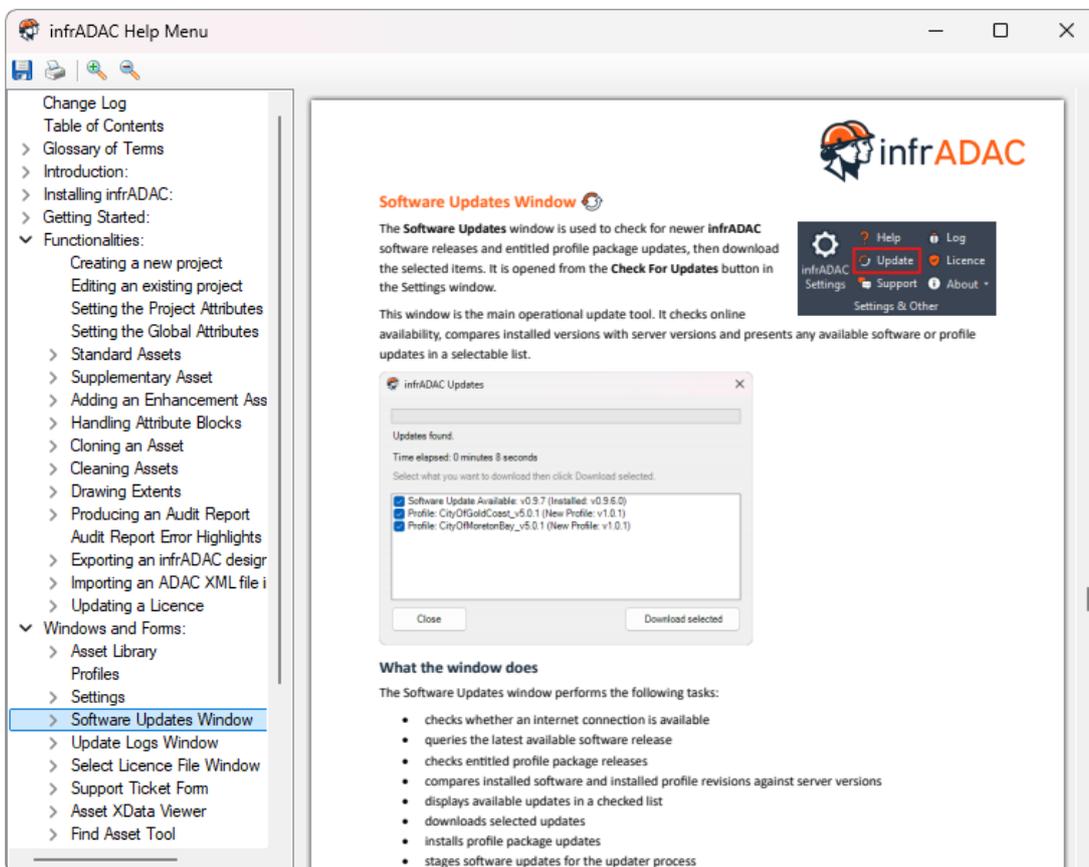
The Help Window is intended to give users a quick reference point for procedures, settings and workflow guidance. Rather than opening an external file separately, the manual is presented directly inside infrADAC in a dedicated viewing window.



What the Help Window displays

When opened, the Help Window shows the current infrADAC user manual in PDF format. The manual is displayed in a built in viewer within the program so it can be read alongside normal work.

The Help Window includes the full manual content, not just a short help summary. This means it can be used to access detailed instructions, step by step guidance and reference information for the features available in infrADAC



Opening the Help Window

The Help Window can be opened from within infrADAC whenever the manual needs to be viewed.

In some situations, the Help Window may open directly to a particular section of the manual. This allows the user to be taken straight to the most relevant topic rather than needing to search for it manually.

This behaviour is especially useful when help is being opened in response to a specific task, form or workflow in the program.



Layout and navigation

The Help Window is designed to make the manual easy to read and navigate. The document opens in a PDF viewing area and includes standard navigation tools.

The window provides:

- a document viewing pane for reading the manual
- a toolbar for standard PDF viewing actions
- a bookmarks pane for moving quickly between sections

The manual is shown in a fitted viewing mode so that page width is displayed clearly within the window. This improves readability and reduces the need for constant manual zoom adjustment.

Using bookmarks

One of the main features of the Help Window is the **bookmarks** panel. This allows the manual to be navigated by section heading rather than only by scrolling page by page.

Bookmarks can be used to move directly to major headings and subsections in the manual. This makes it easier to locate information quickly, particularly when the document becomes longer over time.

If the Help Window is opened for a specific help topic, the viewer may automatically jump to the matching bookmarked section of the manual. When this occurs, the user is taken directly to the relevant part of the document as soon as the Help Window opens.

Toolbar functions

The Help Window includes a standard PDF toolbar to support normal document viewing. This provides familiar controls for moving through the manual and adjusting the view as needed.

Depending on the viewer controls available, this may include actions such as:

- moving between pages
- changing the zoom level
- fitting the document to the window width
- showing or hiding the bookmarks panel

These controls are intended to make the manual easier to use without requiring the document to be opened in a separate PDF application.

Reading the manual in the Help Window

The Help Window is intended for reference and guidance only. It allows the user to read instructions while remaining inside **infrADAC**, which is useful when following a process step by step.

Because the manual is displayed internally, users can move between the manual and the program more easily than if they were switching between separate applications.

This is particularly useful when:

- checking how to use a form or feature
- confirming the meaning of an option or setting
- following a documented workflow while working in a drawing



About Window

The 'ADACabout'  command opens the **infrADAC Disclaimer** window.

This window displays the program disclaimer and general information about ADAC and **infrADAC**. It is used as an information and acknowledgement form rather than a working window.



What the window shows

The disclaimer text explains what ADAC is, what **infrADAC** is used for, and that data accuracy depends on user input, supplied data and selected settings. It also directs users to the built in support system in Settings or to the support email address for assistance.

Accepting the disclaimer

The window is closed by selecting **Accept**.

When **Accept** is pressed:

- the acceptance is recorded by **infrADAC**
- the form closes

This acceptance is stored so the software can recognise that the disclaimer has been accepted.

