МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп’ютерної та програмної інженерії
Кафедра комп’ютерних систем та мереж

“ДОПУСТИТИ ДО ЗАХИСТУ”
Завідувач кафедри

_________________ Жуков І.А.

“_____”____________2020 р.

ДИПЛОМНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

випускника освітнього ступеня “МАГІСТР”

спеціальності 123 «Комп’ютерна інженерія»
освітньо-професійної програми «Комп’ютерні системи та мережі»

на тему: “**Комп’ютерна система інформаційного моделювання інженерного обладнання будівель і споруд з використанням cloud-технологій”**

Виконавець: ___________________________________ Крамаренко І.П.

Керівник: _____________________________________ Кудренко С.О.

Нормоконтролер: _______________________________ Надточій В.І.

Засвідчую, що у дипломній роботі
немає запозичень з праць інших
авторів без відповідних посилань
______________ Крамаренко І.П.

Київ 2020

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
NATIONAL AVIATION UNIVERSITY
Faculty of Cybersecurity, Computer and Software Engineering
Computer Systems and Networks Department

"PERMISSION TO DEFEND GRANTED"

The Head of the Department

___________________ Zhukov I.A.

"_____"____________2020

MASTER'S DEGREE THESIS

(EXPLANATORY NOTE)

Specialty: 123 Computer Engineering
Educational-Professional Program: Computer Systems and Networks

Topic: **"Computer system for information modeling of engineering equipment of buildings and structures using cloud technologies"**

Completed by: ____________________________________ Kramarenko I.P.

Supervisor: ____ Kudrenko S.O.

Standards Inspector: _________________________________ Nadtochii V.I.

Kyiv 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп’ютерної та програмної інженерії
Кафедра комп’ютерних систем та мереж
Освітній ступінь: «Магістр»
Спеціальність: 123 «Комп’ютерна інженерія»
Освітньо-професійна програма: «Комп’ютерні системи та мережі»

“ЗАТВЕРДЖУЮ”
Завідувач кафедри
__________________ Жуков І.А.
“_____”_______________2020 р.

ЗАВДАННЯ
на виконання дипломної роботи
Крамаренка Івана Петровича
(прізвище, ім'я та по-батькові випускника в родовому відмінку)

1. Тема дипломної роботи: «Комп'ютерна система інформаційного моделювання інженерного обладнання будівель і споруд з використанням cloud-технологій» затверджена наказом ректора від 25.09.2020 р. № 1793/ст

2. Термін виконання роботи (проекту): з 1 жовтня 2020 р. до 25 грудня 2020 р.

3. Вихідні дані до роботи (проекту): *інфраструктура на базі Amazon Web Services та програми для проектування Autodesk Revit, що включає в себе BIM-моделі, проекти, плагін Revit і сервіси, такі як Simple Storage Service, Relational Database Service, Virtual Private Cloud, Identity and Access Management, Elastic Compute Cloud, Elastic Beanstalk.*

4. Зміст пояснювальної записки: *Введення в BIM, програмування систем моделювання для інженерного обладнання, розробка інфраструктури для контролю стандартів, огляд Autodesk .NET API та створення плагінів системи моделювання інженерного обладнання.*

5. Перелік обов'язкового графічного (ілюстративного) матеріалу: *Графічні матеріали результатів дослідження надати у вигляді презентації у форматах .ppt, .pdf.*

6. Календарний план-графік

№ пор.	Завдання	Термін Виконання	Підпис керівника
1	Узгодити технічне завдання з керівником дипломної роботи.	1.10.20-8.10.20	
2	Виконати пошук та вивчення науково-технічної літератури за темою роботи.	9.10.20-24.10.20	
3	Опрацювати теоретичний матеріал щодо стану науково-технічної проблеми.	25.10.20-27.10.20	
4	Проаналізувати відомі підходи та методи розв’язку проблеми, розробити аналітичну модель готового рішення.	28.10.20-04.11.20	
5	Розробити імітаційну інфраструктуру для Revit і AWS та протестувати функціонал.	05.11.20-15.12.20	
6	Порівняти ефективність аналітичного та імітаційного моделювання.	16.12.20	
7	Виконати аналіз результатів готової інфраструктури, розробити рекомендації щодо подальшої розробки та оформити пояснювальну записку.	06.12.20-12.12.20	
8	Оформити графічну частину записки та подати матеріали роботи на антиплагіатну перевірку матеріалів.	13.12.20-14.12.20	
9	Отримати рецензію та відгук керівника. Надати матеріали роботи на кафедру.	15.12.20 18.12.20	

7. Дата видачі завдання: “1” жовтня 2020 р.

Керівник дипломної роботи _________________________ Кудренко С.О.
(підпис керівника)

Завдання прийняв до виконання _____________________ Крамаренко І.П.

NATIONAL AVIATION UNIVERSITY

Faculty of Cybersecurity, Computer and Software Engineering
Department: Computer Systems and Networks
Educational Degree: "Master"
Specialty: 123 "Computer Engineering"
Educational-Professional Program: "Computer Systems and Networks"

"APPROVED BY"
The Head of the Department
__________________Zhukov I.A.
"_____"_______________2020 p.

Graduate Student's Degree Thesis Assignment

Kramarenko Ivan Petrovich

1. Thesis topic: "**Computer system for information modeling of engineering equipment of buildings and structures using cloud technologies**"
approved by the Rector's order of 25.09.2020 р. № 1793/ст

2. Thesis to be completed between from 01.10.2020. to 25.12.2020.

3. Initial data for the project (thesis): *Amazon Web Services infrastructure and Autodesk Revit design software, which includes BIM models, projects, the Revit plugin, and services such as Simple Storage Service, Relational Database Service, Virtual Private Cloud, Identity and Access Management, Elastic Compute Cloud, Elastic Beanstalk.*

4. Contents of the explanatory note: *Introduction to BIM, modeling systems programming for engineering equipment, development of infrastructure for standards control, review of Autodesk .NET API and plug-ins creation of system of engineering equipment modeling.*

5. The list of mandatory graphic materials: *Graphic materials of research results should be provided in the form of a presentation in .ppt, .pdf formats.*

6. Timetable

#	Completion stages of Degree Project (Thesis)	Stage Completion Dates	Signature of the supervisor
1	Technical task coordination with the supervisor	1.10.20-8.10.20	
2	Selection and study scientific literature on the topic of master's degree thesis	9.10.20-24.10.20	
3	Work through theoretical material on the state of the scientific and technical problem.	25.10.20-27.10.20	
4	Analyze the known approaches and methods of problem solving, develop an analytical model of the finished solution.	28.10.20-04.11.20	
5	Develop a simulation infrastructure for Revit and AWS and test the functionality.	05.11.20-15.12.20	
6	Compare the effectiveness of analytical and simulation modeling.	16.12.20	
7	Analyze the results of the finished infrastructure, develop recommendations for further development and issue an explanatory note.	06.12.20-12.12.20	
8	Make a graphic part of the note and submit the materials of the work for anti-plagiarism.	13.12.20-14.12.20	
9	Get a review and feedback from the manager. Provide materials for the department.	15.12.20- 18.12.20	

7. Assignment issue date: 1.10.2020

Diploma Thesis Supervisor ________________________________ Kudrenko S.O.
(Signature)

Assignment accepted for completion _________________________ Kramarenko I.P.
(Student's Signature)

ABSTRACT

The Explanatory Note to the Magistry's Degree Thesis "Computer system of information modeling of engineering equipment of buildings and structures using cloud technologies": 112 pages, 69 figures, 21 references.

Purpose of the Magistry's Degree Thesis: analyze modern BIM development problems, solve them by creating the system of system of information modeling of engineering equipment of buildings and structures using cloud services.

Main Tasks: to analyze the process of modeling of engineering equipment of buildings and structures and find ways to simplify it.

Designing object of research: processes automatization for modeling of engineering equipment of buildings and structures with the help of modern cloud technologies.

Subject of research: computer system for information modeling of engineering equipment of buildings and structure using cloud technologies.

Practical usage: the system increases designing efficiency, simplifies and speeds up a work of engineers in modeling companies of engineering equipment, buildings, and structures, helps complying strict project standards and coordinate it.

Main metrics and results: the results of the Master's Degree Thesis can be used in designing or engineering companies where modeling process of engineering equipment, buildings or structures has a significant role.

INFORMATION MODELLING, REVIT, PLUG-IN, .NET PROGRAMMING, AUTOCAD, AUTODESK, BIM TECHNOLOGY, AMAZON CLOUD TECHNOLOGIES.

CONTENT

LIST OF SYMBOLS, ABBREVEATIONS, TERMS

AWS – Amazon Web Services

EBS – Elastic Block Store

EC2 – Elastic Compute Cloud

AMI – Amazon Machine Image

EFS – Elastic File System

VPC – Virtual Private Cloud

IAM – Identity and Access Management

MPI – Message Passing Interface

PPS – Packet Per Second

HPC – High Performance Computing Clusters

SSL – Secure Sockets Layer

REST – Representational State Transfer

API – Application Programming Interface

IFC – Industry Foundation Classes

ISO – International Organization for Standardization

BFC – BIM Collaboration Format

PHP – Hypertext Preprocessor

HTTP – Hyper Text Transfer Protocol

SDK – Software Development Kit

MMAP – Memory-mapping

NFS – Network File System

DBA – Database Administrators

DBC – Drop Clean Buffers

CGI – Common Gateway Interface

BIM – Building Information Modeling

CAD – Computer-aided Design

IOT – Internet of Things

INTRODUCTION

Actuality of theme – public and private sector clients are increasingly demanding their projects are realized on a BIM platform. Despite some countries adopting it at a faster rate than others, there is consensus that the world's built-environment will benefit dramatically from the potential inherent in BIM to improve its social and environmental performance. BIM is key in the pressing search for more sustainable solutions and is already moving towards becoming an industry wide approach at a global level.

One of the areas that have received heightened attention recently is the automation of various tasks in the field of engineering and architecture. More often it is possible to encounter this as generative design. It is chosen to explore the huge power of automation for this research work as it is not limited to just BIM modelling or geometry, applying automation for various tasks from geometry generation to data, parameter management, simplifying complex or time-consuming tasks, such as creating and sorting schedules in Revit, etc.

Learning the first time any CAD systems or moving to a new level of expertise or changing job responsibilities, it is not unusual to question the value of programming. After all, if you are an CAD systems user, your job is to produce drawings, not to make programs. Sometimes CAD managers are responsible for creating programs to improve workflow and quality. And in rare cases, a company will hire a programmer to automate some aspect of its work.

Most of tools from current diploma project are dedicated towards either software function extension (in this case it is Revit) or automation of repetitive/complex tasks. Tools created to automate certain processes include numbering reinforcement, elements, inserting custom parameters according to element types or a set of other parameters, generating necessary views of elements. When talking about the extension of Revit capabilities, these usually include placement of elements at specified world coordinates, the complex distribution of reinforcement bars, modelling of complicated geometry and etc.

Today's information modeling companies are the present and future of building engineering, as they have extensive capabilities and high detail in the final product. But even given these factors, there are plenty of problems and things to improve. Primary among them are routine tasks, human errors, and adherence to strict standards. These are the problems this master's project aims to solve, as there are almost no ready-made solutions on the market.

In this paper, the first section introduces BIM, its relevance and necessity in modern design, broad possibilities, a springboard for use, methodology and development, as well as a view of it all from the management side. The first section also describes possible software tools using building information modeling and gives examples of automation of some of them.

The second section is devoted to a thorough analysis of information modeling processes, their problems, and options for improvement using the extensive capabilities of programming in tandem with modern Amazon cloud technologies. A comparison of actual ways of automation, their possibilities and further development for choosing the most modern and functional way will be introduced.

The third practical section focuses on the direct step-by-step implementation of a broad infrastructure that includes several Revit plug-ins, a database, a user rights management system, cloud storage, virtual machines for query processing, and many other cloud service components to create a self-contained and useful system for BIM professionals.

Research methods – to understand the problem itself there was implemented numerous experiments including real engineers and architectures. The experiment was based on the time consumed during similar projects and the average amount of mistakes made with implemented tool and without. Moreover, the feedback of those specialists has been accounted to make a final product more comfortable and well-to-do not only from the sight of the computer engineer but from the sight of BIM specialists.

The purpose of the thesis – explore modern BIM development problems, solve them by developing the system of system of information modeling of engineering equipment of buildings and structures using AWS cloud services.

Object of research – automatization of processes information modeling of engineering equipment of buildings and follow concrete project standards.

Subject of research – computer system for automatization of processes information modeling of engineering equipment of buildings and structures using project standards and modern cloud services.

The practical significance of the results – As a result of research through surveys of current BIM engineers and specialists, the following product is highly recommended to engineers of any direction (E/H/V/AC), as well as to architects and BIM coordinators, because implemented infrastructure greatly simplifies the design and compliance with standards. The evidence of this is the successful practical application of this product by many companies in Ukraine, Switzerland, and Germany. Thanks to a wide user base and the feedback mechanism, the entire infrastructure is actively improving and evolving over time.

Personal contribution – software engineer Roman Lavrov initiated this idea by implementing a small plugin for Revit application that speeds up the routine process of designers. It was this plugin that became the foundation for further development, which led to the established infrastructure, which includes cloud services and several different plugins for BIM specialists.

Publications – the following theme with its key problems, features and implementation field is described in the following publication:

Problems of informatization and management [Electronic resource]. – National Aviation University, 2020. – № 63. – p. 49. – Access mode: http://jrnl.nau.edu.ua/index.php/PIU/article/view/15000. - 24.09.20. – Title from the screen.

PART 1

BUILDING INFORMATION MODEL PROCESSES AND WORKFLOW STRUCTURE

1.1. Application areas and necessity of Building Information Modeling

Building Information Modeling (BIM) is a digital information management approach being adopted by the construction industry to improve productivity and quality in building and infrastructure projects, reduce financial losses during construction, and provide a basis for developing future services. At its core is 3-D modelling with embedded data that can be shared by and amongst all project partners, at all stages of a project from design through to maintenance.

Each partner retains responsibility for their own data but project managers are able to use BIM as one “agreed source of truth” for their decision-making. This means a more efficient use of resources, more effective communications and therefore collaboration within the partnership, greater flexibility and improved long-term planning – the benefits of which accrue to client, project manager and contractor.

Public and private sector clients are increasingly demanding their projects are realized on a BIM platform. Despite some countries adopting it at a faster rate than others, there is consensus that the world’s built-environment will benefit dramatically from the potential inherent in BIM to improve its social and environmental performance. BIM is key in the pressing search for more sustainable solutions and is already moving towards becoming an industry wide approach at a global level [1].

It is possible to indicate the following application areas of BIM:

- the real-time resource location – productivity and safety can be optimized through integration of a BIM model with Radio Frequency Identication (RFID) technology or the barcode-based system;
- digital data management system that allows for real-time storing, finding and sharing these data;
- planning renovation or retrofit and developing a feasibility study for these works;

- activities related to the operation and maintenance of the building;
- energy analysis and simulation, control of electronic systems in the building management phase as part of Facility Management;
- health and safety management during construction and operation.

1.1.1 Modern Building Information Modeling

Recent advances in, for example, Internet of Things (IoT), remote sensors, information preparing and examination, and Building Information Modeling (BIM) can possibly change how we communicate with the fabricated climate and improve the experience for end clients and specialist co-ops. The IoT gadgets and sensors are progressively being sent in the fabricated climate and modern applications.

The quantity of associated gadgets has just overwhelmed associated individuals and are assessed to associate with 9 billion. The sensor hubs are being sent in different application regions, for example, the mechanical, transportation, wellbeing, and prosperity, building robotization, car and retail. The quantity of sensor establishment is expanding at a remarkable rate and a few assessments recommend that there will associate with 50 billion associated gadgets by 2020.

BIM is the current headliner in the development business. In spite of the fact that the innovation has been around for about 10 years, a great deal of buzz has been made about BIM in the field for as long as two years. BIM is the cycle spreading over the age and the executives of the physical and utilitarian data of a venture. The yield of the cycle is the thing that we allude to as BIMs or building data models which are at last advanced records that depict each part of the task and backing dynamic all through a venture cycle. It has been felt that BIM is just 3D demonstrating, yet it really includes more than that [2].

1.1.2 Project management and BIM

The implementation of BIM as a methodology gives the best result by far when the quality of the information deliverables is improved and when the collaboration is enhanced in a structured way and integrated into the project management process [3].

It is essential that the project management process remains the same as always, with all necessary underlying knowledge and experience. BIM brings additional tools, metadata and concepts to the same fundamental traditional processes of design, planning, rendering, execution, communication and coordination. This naturally leads to improvements in reducing financial losses, saving time, improving quality and mitigating risks.

The BIM approach supports the agile and lean principles for better planning of both inputs and deliverables. It improves communication and interoperability between different stakeholders and different project phases and leads to an integrated approach for changes and optimizations.

To some extent, project managers have been left out of the mainstream action on BIM – partly due to lack of engagement on their part and partly because the frontend use of BIM in design has been in the limelight. This situation needs to change – effective project management is critical to ensuring successful BIM implementation and successful project delivery.

Why is the role of project managers on BIM projects so important? The answer lies in the very definition of BIM. From the definition, one can discern key themes of "collaboration," "coordination," "communication," "exchange" and "collation" that are relevant to answering questions about the impact of effective project management in BIM implementation. As per the RICS APC Pathway Guide on Project Management, "Project managers occupy a central role in the development process driving successful completion of projects." It is well established that effective project management contributes to project success. This requires the roles and responsibilities of the project manager to be clearly defined. In today's context, when the use of BIM is likely to become mainstream, this aspect still holds true – rather than diminishing the role of the project manager, BIM actually enhances and sharpens it [3].

Fundamentally, the key role every project manager plays on projects is that of bringing together a disparate group of experts and integrating their knowledge into a buildable design that meets cost, quality, safety, sustainability, and schedule objectives and requirements, and generates value for the project sponsor, project team members, and end-users. The project manager plays a key role as the "Project Integrator", being responsible for the integration across the temporal dimension of the project (the various project lifecycle stages) as well as across the dimensions of organisations, processes, and information. The ultimate success of a project depends on how effective the project manager can be as an integrator across these dimensions. With BIM as a fundamental enabler for effective integration, it is crucial for project managers to understand how to harness and use it for their projects.

1.1.3 BIM implementation from a project manager perspective

The four major criteria of any project – cost, time, quality and risk – are the main parameters that quantify the success of BIM implementation. From an engineering consultant and project manager perspective, effective implementation of BIM means primarily focusing on using BIM as a tool for staying within the planned budget and timeframe, without clashes and problems in design, unplanned works, reworks, claims, unrealistic deadlines, etc. [3].

BIM is facilitating project management and quality control through processes and tools. Automated and semi-automated quality control procedures are available using BIM tools for different BIM uses in different project stages.

It is essential that project managers on BIM projects fully understand the BIM process and know how to use the model to track progress as it is different from traditional projects. It is, for example, impossible to track progress using numbers of completed drawings on a BIM project.

Having project managers in line with BIM, means to integrate its principles and technology into various project processes and to utilize the enhanced level of information for decision-making, quality control and effective risk mitigation.

Effective BIM implementation requires a lean, start-with-the-end-in-mind approach, which involves project requirements, delivery and contract type, the number of stakeholders and existing infrastructure and buildings. Information which is entered and not used is considered waste, but too little information can inhibit some later BIM uses and analyses.

The BIM model itself, the “digital twin”, is the basis of information for most of the BIM uses derived from it, and since this is the case, the information requirements for it should be precisely defined having in mind all subsequent uses of the data, further input, process and software requirements, etc.

The defined information requirements for each of the BIM deliverables should take into account the software output interoperability between deliverables and should ensure the outputs from one procedure can be utilized as inputs and references of another – a piece of information entered once should propagate through the process and be used to the highest extent.

The process of defining information requirements and information handshakes should apply the principles of systems engineering. This means that information is not lost to another participant in the project, and that collaboration is enhanced not only between immediate co-workers and co-participants, but between stakeholders who will not have direct contact with each other on the project (e.g. designers and facility managers on some projects which follow a waterfall design-bid-build-operate method). This dramatically improves the quality of deliverables, making obsolete documents and miscommunication highly unlikely, with communication being enabled through the BIM model itself [3].

When project management processes are linked with BIM uses, deliverables and information management, the true value of the BIM model as a single source of truth can be fully harnessed.

The slight initial expense in the transition process is usually covered through mitigated risks and saved unplanned work. Although one should bear in mind that these types of expenses are hard to estimate and even harder to prove, thus sometimes making calculating return on investment difficult.

Risk management is a standard responsibility of project managers, as well as a concern for investors and contractors, which can also benefit from BIM uses. Much of the risk related to the quality and synchronization of design, obsolete or imprecise quantities, undefined or out-of-scope works, unrealistic schedules, unplanned rework due to collisions, bad cost estimates, can all be severely reduced and mitigated with proper implementation of BIM as a tool for these estimates or a tool for quality control. This can result in more realistic schedules and cost estimates, less unpredictability, and even less risk margins with bidding and tenders.

1.1.4 BIM Processes

ISO 19650 clarifies measures all in all terms. Be that as it may, there are some major questions for the designing and development periods of BIM-empowered activities of which counseling specialists should know [3]:

- the characterized measure should consider advancements in the task and any progressions this makes to the expectations, and so on the part of the BIM chief isn't characterized in the ISO 19650 and depends of the nation, the agreement, the organizations;
- the function of the BIM administrator in an undertaking group fluctuates across organizations, It must be considered as a job, related to the delegated party in the agreement, including the cycles of BIM the executives with the obligation of the BEP, the data the board, the data necessities the board, the data work process the board and the data interfaces the executives. Indeed, on the off chance that it's anything but a full-time movement, an individual from the venture supervisory crew must be named for these obligations;
- the significance of characterizing the cycle for trading data between all partners.

The recurrence and nature of the data trade and sharing is critical to the achievement of a BIM venture and should be aligned with the normal expectations. The BIM model, the "advanced twin" , is a consolidation of various models and goes about as a solitary wellspring of truth for all data got from it [3].

To guarantee quick intermingling of the BIM model towards a last plan that conforms to prerequisites, and eventually with the truth of the resource, it is fitting to just share the data that is important to be shared at that specific second, with different partners. The BIM management is to organize the following tasks, and their implementation under the BEP:

- to merge all discipline models;
- to check the compliance of the models according to the BEP or other agreement;
- to check the evolution of the models and their compliance with the scope of work;
- to check for clashes between the models.

After defining, in the BEP, the actions needed under the design contract, together with the project manager or the steering team, the BIM Management needs to impart the concluded activities to all partners. The cycle rehashes itself on a concurred recurrence until agreement or objective per venture stage has been reached and the plan can be distributed.

Because of the abovementioned, the BIM cycle makes more noteworthy correspondence and perceivability around venture progress for all partners thus bolsters venture the executives in a positive manner. Nonetheless, it requires an attitude of sharing and straightforwardness that isn't generally normal in conventional tasks, nor among numerous partners across the entire development area.

It merits referencing that this cycle likewise requires a lot nearer association of the customer during all undertaking stages, and explicitly with respect to endorsement after each stage which is essential for having the option to unite towards a last plan.

During the operational or upkeep stage, the measure of cooperation with various partners will typically be not exactly during the task stage, so the above is fitting just on a need-to-have premise. This stage is getting increasingly more about enhancing the BIM model with extra data and utilizing this data for creative methodologies (Internet of Things, Smart Buildings, Smart Cities, and so on) and support or for operational purposes.

1.1.5 Exchanging and sharing information in a BIM process

To facilitate the use of BIM and the exchange or sharing of digital information between clients and suppliers across the project and asset lifecycle, regulatory, procurement and legal aspects should be considered to clarify the terms relating to [4]:

- hierarchy of the information
- intellectual property ownership
- obligations and liabilities of suppliers
- purpose of information exchanges
- roles and responsibilities for information management

BIM information can be shared via ‘containers’ according to the ISO 19650 fundamental principles of container-based collaborative working to be achieved [3]. That is collaborative working across an asset or project team using containers for sharing asset or project information. An example of a container-based collaborative way of working is in the open standardized “COINS 9” container, using an information delivery manual (IDM) to describe the exchange requirements.

1.1.6 Common Data Environment

During the lifecycle of an asset, there is a continuous flow of information. The CDE captures, controls, manages and shares this information (called “metadata” by ISO 19650) throughout [3].

The evolving 3D and BIM software can carry out tasks related to:

- document management, including revision control;
- access rights to information;
- workflow, log books, audit trail, progress control, cost control;
- connection to corporate systems, e.g. for financial data;
- project communication, e.g. with communication through email, pdf files, etc.
- BIM tools like viewers, clash detection and code checking, etc. (also supports communication between stakeholders);
- up-to-date information take-off;
- adding separate graphical and non-graphical information;
- maintenance and facility management.

The transition from file-based to model-based project implementation has led to a change of work methodology and interaction patterns. The models are 10 to 100 times larger than the previous model files, and calculations are performed directly in the models. This has resulted in new requirements for storage capacity, file sharing and accounting capacity. The requirements for infrastructure, hardware and software are increasing exponentially and software vendors have changed their strategies to comply with the new requirements. Cloud-based solutions are increasing storage capacity, distributing software and databases (weather and climate data, IoT) and interacting with projects across multiple companies without having to open-up.

Cloud applications are owned by software vendors and are strongly linked to their platform. The number of places where the data are stored therefore increases with the number of applications and their cloud-based solutions that are used in a project. Not every application is flexible in including formats for which vendors do not have ownership, and many solutions cannot use storage sites for which they do not have full control. An open format for CDE, the use of open standards, and linked information applications, based on international open standards, via open accessible “platforms”, while taking care of data security, data protection as well as protection against the misuse of data, will help to prevent the issue described above.

1.1.7 Reasons to use BIM

BIM is a relatively new technology especially in the construction sector, an industry typically slow to adapt to change. BIM proponents claim that in the near future, it will offer a lot of value in terms of:

- improving visualization;
- improving productivity via easy information retrieval;
- increasing coordination of construction documents;
- linking of vital information such as vendors for specific materials, the location of details and quantities required for tendering;
- increasing speed of delivery;
- reducing overall costs.

1.2. BIM deliverables – reusability

BIM deliverables, inputs and their respective structures, should be implemented as stated in ISO 19650, following the information lifecycle concept and setting information requirements with information delivery in mind. This means every participant in the supply chain should adhere to the information handover principles for the BIM model and related deliverables, to induce not only horizontal collaboration within a single project phase or set of participants, but also to enable reuse of this data for participants further along the project pipeline. This is especially true with the "single source of truth" concept of the BIM model, and should be further amplified with the CDE and proper definition of project processes [3].

More specifically, BIM deliverables are to be defined in a lean way – each deliverable should have a clear task at improving the main project features – cost, time, quality and risk. Deliverables should be able to be used and re-used throughout the project lifecycle, avoiding the need for re-entering (essentially waste in lean terminology) information that was already present in a way but is unusable due to technical (software, format etc.) or legal (rights, availability) restrictions. Each deliverable should be structured in line with existing project management processes, with clear quality control and assurance procedures, and workflow instructions on where it will be reused later in the lifecycle and what will be the necessary information handover and requirements for it.

In all projects, especially larger ones with multiple disciplines and numerous stakeholders, it is crucial to enable interoperability by ensuring that all results and useful information can be opened, read, reviewed and further used by any entity involved. There is an increasing number of new software features and capabilities. All information should be prepared for future, unanticipated uses.

This is further exemplified through the following figure where it is shown that (fig. 1.1), in a traditional setting, significant information about the project is lost between different phases, and this is true for between different stakeholders as well.

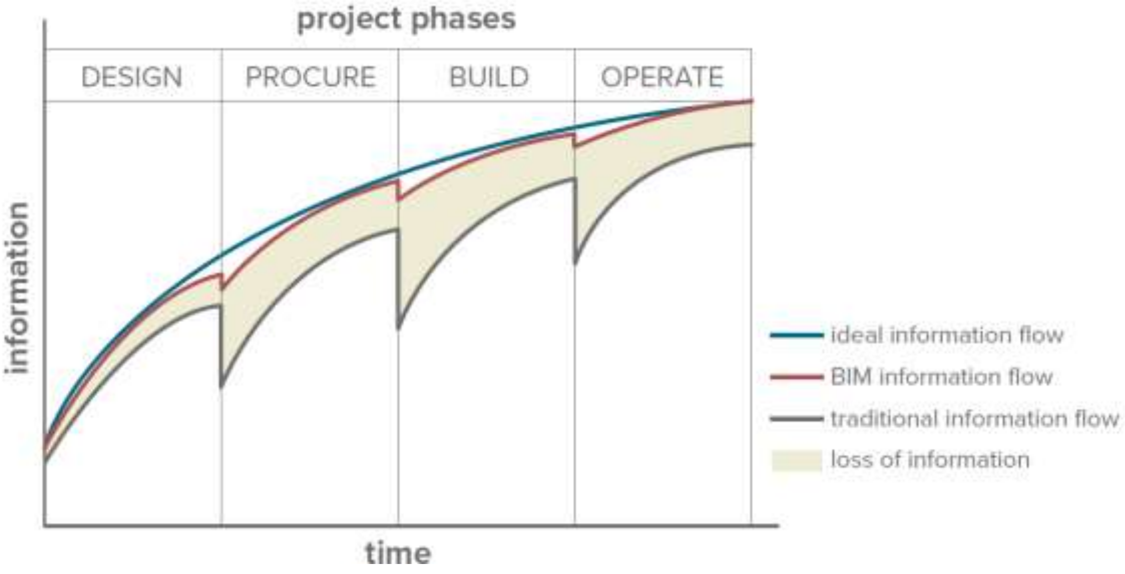


Fig.1.1. Knowledge base throughout the project lifecycle

With an agreed source of knowledge in the form of a BIM model, information is better communicated and remains within the knowledge base even after stakeholders who have implemented this knowledge have finished with their involvement in the project [3].

The interoperability should be available for both the baseline data (BIM model, through the ISO 16739:2013 Industry Foundation Classes standard) and analyses results (clash detection, cost simulations BIM5D, construction simulations BIM4D).

Current open formats should be extensively utilized, as they will be the basis for most open formats in the future (most of them have existed for many years and have had several iterations and versions, gaining popularity and user base along the way):

- Industry Foundation Classes (IFC), as well as the newer IFC 4 with 4D/5D support and oncoming version IFC 5 with infrastructure support (with buildingSMART International being the main organization working on this format);
- BCF (Open BIM Collaboration Format) – for clash detection and viewpoints (buildingSMART);
- CityGML – for digital models of cities and urban areas (Open Geospatial Consortium OGC);
- Other formats in the openBIM initiative for further BIM analyses, including QTO, time schedules, cost estimates, operations and maintenance data, sensory data etc.

These formats are currently either in development or in the planning and definition phases and should be supported officially through standardization bodies, buildingSMART and OGC.

Interoperability definitions should be available for any project container utilized for collaboration. They allow for having deliverables which are independent from software vendors. In order to ensure the required interoperability, it is recommended to use software that supports open standardized formats and which are certified by relevant organizations, such as buildingSMART IFC Certification [4].

The Project Execution Plan, the BIM Execution Plan, and project contracts, should specify the desired open formats from the beginning, as the planned end-results and deliverables should be readable for the client and any future potential participant, without having to incur additional costs.

This means using all open formats available for defining the deliverables, along with sample files for evaluating readability and usability of the end deliverables. If open formats are not readily available for a set of analyses, the specific format should be mutually agreed, and other variants of the format should be explored (e.g. other proprietary formats which can be read using an application or software development kit (SDK) as a set of software development tools that allows the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform).

Under the BEP approval process, all sample files should be evaluated by the client and approved officially prior to a specific phase of the project. The pre-approval of all formats vastly reduces interoperability risks and synchronization and conversion time and improves usability of future deliverables.

The CDE vendor should be approached for an open source format as well, for potential migration and backup of versioned project data. This format should be defined and further developed, which could be done as a proposal or in collaboration with some of the openBIM initiatives.

All open formats and standards should be a part of the client's information requirements, especially for the asset information requirements for operations and maintenance use.

1.3. Software tools

However, BIM, along with huge data stores (Big Data) being accessible through the world wide web, changes the world for suppliers of ideas and services like architects and engineers. In the end, it could be possible that their ideas find their way easily into new projects – including all connected data like calculations, descriptions of materials and so forth. It is important to think about worldwide copyrights and Big Data based services that can protect intellectual property rights. The evolution of BIM has been strongly supported by software and working in a BIM process is simply not possible without it in some way, whether it is limited to the creation of 3D models or whether it concerns the full cooperation of all stakeholders in a Common Data Environment (CDE) [8].

Since BIM is all about evolving towards a digital twin, the "single source of truth", software vendors become an important and powerful supplier in a project. Digitization is more and more part of our global culture and is evolving and changing very fast. The lifespan of assets, however, is much longer than that of a certain software or versions of it and this fact alone can create issues of accessibility and ownership of data over time. It is advisable to thoroughly think through, and fix in contractual BIM agreements, the guaranteed access, readability, security, back-up, legally required physical copies of the project, access and property rights, etc. of all data being transferred during each phase and over the expected lifetime of an asset.

Specific attention should be drawn to cloud-based solutions with regard to ownership and the free use that software vendors have over the project data. In projects that make use of full CDE potential his could even concern all communication, financial and planning data. Also, contractual agreements should advisably contain mechanisms on how to deal with partners joining and leaving the project team with regard to software implications. For example, what if the company that established, and controls, the CDE platform leaves the project team?

In projects using multiple software with Cloud solutions, there is a risk that the concept of “Single source of information” may be compromised. The architecture, engineering and construction industry should apply open BIM standards and use open cloud software products. This is also in the interest of the IT-industry because their business case is growing, by a growing international standardized market, for the same tools.

1.4. Selecting programming language for modeling system of engineering equipment

Each expert has over and over been asked which is the 'best' API to use for a 3D improvement venture. The appropriate response has been genuinely predictable, regardless of whether the innovation itself has changed. The decision relies upon various elements, some of which let me attempt to list here in no specific request:

Legacy code base - is there a current heritage code base, and does it function admirably for any customers? Assuming this is the case, are there approaches to communicate in/from the heritage language to the language you intend to receive?

Functionality coverage - what sort of usefulness is needed from the language? Does it require hard calculating abilities, is it about an extravagant or instinctive GUI?

How do these necessities coordinate the capacities of a) the language b) the arrangement of APIs displayed through this specific API for AutoCAD?

Efficiency - identified with the past inquiry - is an issue of execution. Is what you think about the productivity of the language you are thinking about from the proficiency your clients are requesting?

Integration with other technologies - do you have prerequisites for incorporation with different items or benefits, and does the language have insight about the capacity to help such mixes?

Variant help - or do all stages you need to help (both OS and form of AutoCAD) permit you to utilize this language?

Skills - do you and your partners have the experience/capability/certainty to get familiar with the language? How accessible are these aptitudes in the market today - in the event that you needed to enlist somebody to deal with your code, would it be anything but difficult to track down these abilities?

Vendor commitment - or does the vendor (in this case we could consider Microsoft and Autodesk as major technology providers) have a long-term commitment to the technology - not only because of its availability, but also as it develops? One obvious example here is VBA, which - unless things have changed while I was napping - is not provided as a 64-bit version. Which effectively means that there is virtually no long-term commitment on Microsoft's part to this technology, clearly limiting the possible commitments of vendors like Autodesk.

Support - do you have a solid knowledge base for this technology and a skilled, experienced support staff? If you need to get out of the hole quickly, is there someone you can call?

Our current choice would be if the implementation of a "green field" development project, without considering an obsolete codebase, would focus on one or more .NET languages (including a managed code performance profile within AutoCAD, which approaches native C++ as irrelevant for most purposes, combined with the interface features provided by WinForms and - increasingly - WPF).

We could choose ObjectARX (depending on what was needed). It is still true today that custom objects can only be implemented with ObjectARX (i.e. C++). We will carefully consider the alternatives available to me before going down the route of implementing custom objects (we are still working on ways to reduce the currently significant resource investment required to implement the functionality currently available only through custom objects).

Having a homogeneous code base is certainly beneficial: all code is written in a specific language, uses the same tools, and developers can dip in and out of different areas of code without much effort that might be needed. The interaction between own C++ and managed code is quite complete.

1.5. 2D Drawing Automation in AutoCAD using C#

It gives step-by-step instructions to start with Autocad automation process for beginners.

1.5.1 Background of Autocad

Autocad is a software for creating 2D and 3D drawings [7]. Autocad Automation means that, creating drawings automatically in Autocad by using programming. We can draw very big schemes within few seconds using programming. Autocad provides interoperability with other applications, by using Autocad object model, it is possible to create drawings or any type of models. The above diagram shows the idea behind the Autocad Automation. A custom C# application creates drawing in Autocad by calling available commands from it (fig. 1.2).

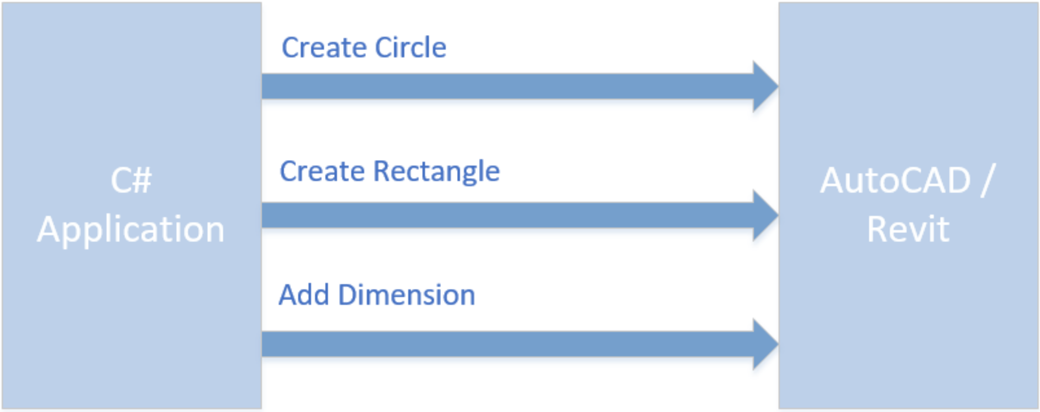


Fig. 1.2. C# calling available commands from Autocad

In this example, it will be created a simple drawing in AutoCAD by a C# application using AutoCAD COM API. Before proceeding further, it is necessary to know the following things: how AutoCAD automation is working, what is AutoCAD object model, how to connect an application with active instance of AutoCAD?

Have a look at the following diagram, which shows the actual working mechanism of AutoCAD Automation (fig. 1.3).

AutoCAD organizes everything as object (OOP), circle, line, rectangle, square, ellipse and all are considered as objects in AutoCAD. Each object in AutoCAD has its own properties and functions.

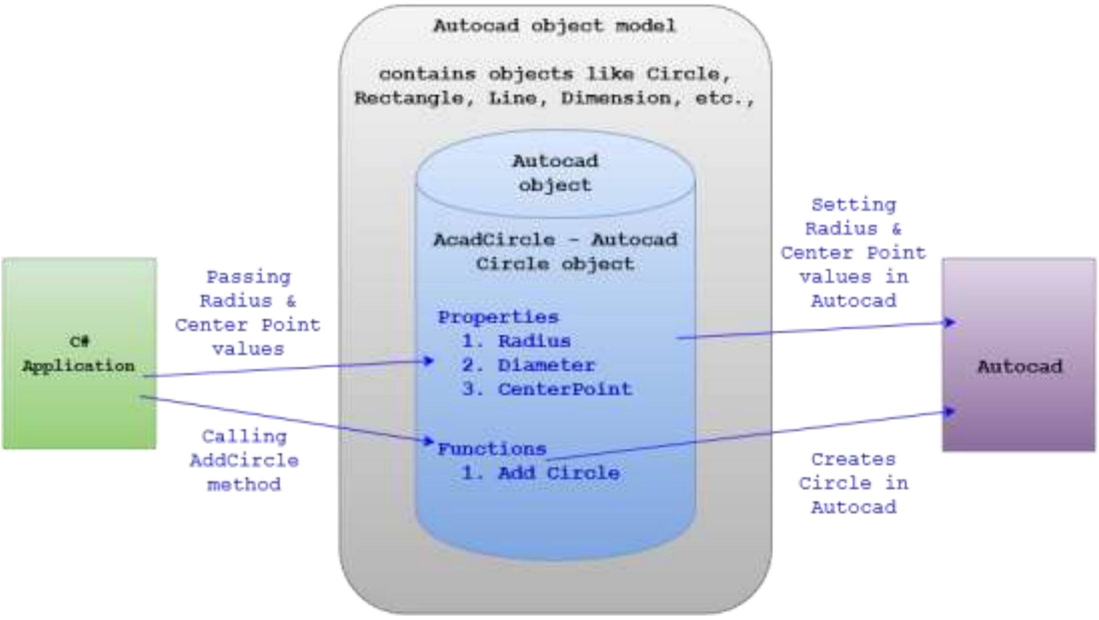

Autocad object model
contains objects like Circle, Rectangle, Line, Dimension, etc.,
Autocad object
AcadCircle - Autocad Circle object
Properties
1. Radius
2. Diameter
3. CenterPoint
Functions
1. Add Circle
C# Application
Passing Radius & Center Point values
Calling AddCircle method
Setting Radius & Center Point values in Autocad
Creates Circle in Autocad
Autocad

Fig. 1.3. Actual working mechanism of Autocad Automation

Autocad presenting these items to the world through Autocad COM API, by utilizing this API in our programming, we can play with the objects of Autocad. In this way, controlling the Autocad objects by means of Autocad COM API is the fundamental thought behind the Autocad Automation [7].

Think about the model from the above chart, The assortment of Autocad items and its association together known as Autocad object model. This model contains the insights concerning all Autocad item's properties and capacities. These articles are accessible in the Autocad COM API. This item model resembles a pool of Autocad objects, we can utilize those items to make attracting Autocad. The chart shows a portion of the properties and elements of the Circle object present in the Autocad COM

API, By calling the AddCircle technique from C# application, the circle will be made in Autocad. By setting the Radius and CenterPoint properties we can control the made circle.

1.6. Use of Revit API

API is the acronym for Application Programming Interface: the way a software programmer can communicate with a software product [7]. For instance, the Revit API is the way programmers can work with Revit, and it establishes what functionality a software programmer can use within Revit (fig. 1.4). Such as the Revit API allows you to write instructions for Revit to execute one after the other.

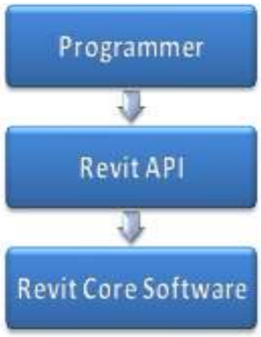


Fig. 1.4. Revit API interconnection

A software plug-in is a type of program module (or file) that adds functionality to a software product, usually in the form of a command automating a task or some customization of the product’s behavior. When you talk about a plug-in for Revit – and you will also hear the term Add-In used for this product – we mean a module containing code that makes use of the Revit API. Revit loads such plug-ins and uses them to adjust its behavior under certain conditions, such as when a particular command is executed by the user of the plug-in.

1.7. Automation of BIM Processes

One of the regions that have gotten elevated consideration as of late is the mechanization of different assignments in the field of designing and engineering.

All the more frequently it is conceivable to experience this as generative plan. It is picked to call it robotization for this exploration function as it isn't restricted to simply BIM displaying or math, applying computerization for different errands from calculation age to information, boundary the executives, streamlining complex or tedious undertakings, for example, making and arranging plans for Revit, and so on [8].

The greater part of devices from current recognition venture are committed towards either programming capacity augmentation (for this situation it is Revit) or robotization of tedious/complex assignments. Instruments made to computerize certain cycles incorporate numbering fortification, components, embeddings custom boundaries as indicated by component types or a bunch of different boundaries, producing vital perspectives on components. When discussing the augmentation of Revit abilities, these typically incorporate situation of components at determined world arranges, the mind boggling dispersion of fortification bars, demonstrating of convoluted calculation and so on Last two gatherings are exceptionally intricate - robotized formats and generative plan apparatuses. Computerized layouts, in our view, require collaboration between an organization predefined formats, boundaries, guidelines and parametric models. For our situation, these consolidate various other secluded instruments into an incorporated framework. Generative plan apparatuses - less perplexing than robotized formats yet complete various complex errands to create BIM models, for example, a device to disseminate certain components, similar to acoustic obstruction dividers, keeping determined ways with predefined rules and with significant levels of flexibility in their factors and boundaries.

Prior to picking explicit undertakings to commit your assets to, recognize early, which venture organizes your organization squanders most energy on, which have the greater part of the redundant assignments and which are generally inclined to human mistakes because of multifaceted nature or different variables. Subsequently attempt to break down the likely reserve funds in time from normalization, parametrization, and robotization in every one of these means. From designers' very own insight, it was discovered the base figure to most precisely speak to the likely investment funds as expected and improvement in quality (fig. 1.5). The cases beneath are not speaking to a particular rates, simply outline possible reserve funds in time comparative with different stages.

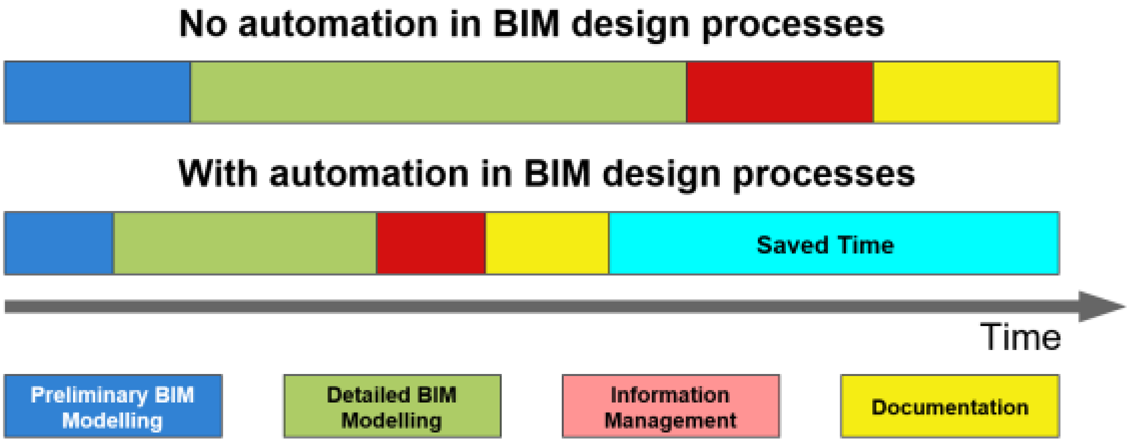


Fig. 1.5. Chart of BIM automatization efficiency

Benefits are to be had regardless of which project stage it is adopted. In our case, we started with BIM modelling as we saw maximum potential there to reduce overall project time. Increased efficiency in model creation should also cascade quickly to other stages. Nevertheless, automation in any of the process stages can make improvements in time consumption and provide a less prone to errors, more efficient workflow to engineers/architects. Company specifics and overall strategy are key in considering which areas should automation be focused on to achieve better results. Each company is different in their own way. Regardless of the area of application, another choice to make before beginning is between initially automating large complex tasks or start with multiple smaller and isolated tasks. The advantage to the first is obviously that it can bring greater benefits but with larger risks in case too much time is consumed developing this tool, the tool does not bring the expected efficiency or the company fails to create one altogether. Therefore, going for multiple smaller automated tasks initially is the better option, as the risks are spread out, in case a few attempts to automate some processes fail, others will succeed. Another significant advantage of taking it more slowly and beginning with simpler automation tools or just isolated problems is the ability to collect usage data from your engineers/architects. This leaves room for improvement further down the road and provides some statistical measurements to evaluate the actual time savings or data for future comparisons [8].

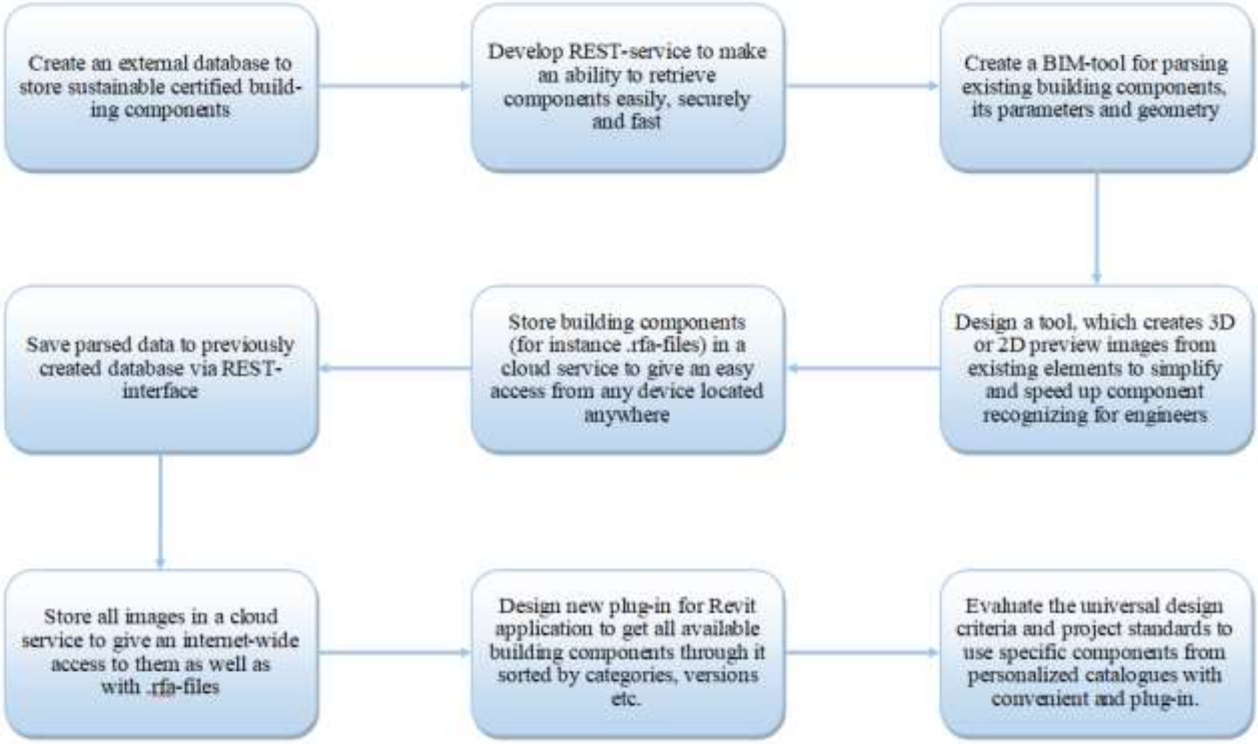
When your company has enough collected data, enough standardized, written down rules, parametric models and tools for generative design, or just to automate some

BIM model management tasks, a whole new world of possibilities opens up. Namely, the world of neural networks, machine learning, or as popularly referred to AI systems.

These are cutting-edge tools, that greatly expand on what is possible to achieve, what is possible to innovate on in our sector. These systems have only recently gained popularity and this has hyped up the sector tremendously.

1.8. Methodology and Model Development

The aim is to develop an automated way in which sustainable design of any building project is accomplished with the help of tool, which speeds up and simplifies creation of BIM object of any building or construction (fig.1.6). This plug-in also gives an opportunity to make the creation process standardized for all engineers involved in the project. Since the proposed tool includes a lot of stages and integrates different applications, as is represented in Figure, the development will be implemented through the following five phases.



Create an external database to store sustainable certified building components
Develop REST-service to make an ability to retrieve components easily, securely and fast
Create a BIM-tool for parsing existing building components, its parameters and geometry
Design a tool, which creates 3D or 2D preview images from existing elements to simplify and speed up component recognizing for engineers
Store building components (for instance .rfa-files) in a cloud service to give an easy access from any device located anywhere
Save parsed data to previously created database via REST-interface
Store all images in a cloud service to give an internet-wide access to them as well as with .rfa-files
Design new plug-in for Revit application to get all available building components through it sorted by categories, versions etc.
Evaluate the universal design criteria and project standards to use specific components from personalized catalogues with convenient and plug-in.

Fig 1.6. Methodology of the Family Browser system implementation

Phase 1. It consists of designing the model’s relational database needed to design sustainable catalogue of building components. A consistent information system depends on the integration between databases, programming languages, and software engineering and its lifecycle incorporates the interrelated technologies of conceptual BIM modeling and database design. The design and development of this database is accomplished in two steps starting by the conceptual modeling with the help of theoretical structure of building components such as family files (.rfa) or templates (.rtf) and ending by the physical implementation. The information related to the green materials is stored in an external database in the form of predefined design families that can be recognized by BIM tool. The separate database is linked to the predefined library of Revit by defining its path and it is loaded every time the BIM tool works. The data related to the families is saved as family files (RFA) or Revit files (RVT), which can be identified by the future BIM tool. Thus, in the external sustainable database, thousands designed families are stored in parameters form. Different types of information such as details about the materials used, suppliers' contact data, assigned keynotes, potential LEED criteria, and assembly codes are stored in the external database.

Phase 2. Phase 2 focuses on creating BIM tool to fit the modularity requirements of the .rfa-file model. The first step is to design and implement a module capable of parsing created families, in BIM tool, and their associated keynotes for components commonly used in projects. The module is linked to the database developed in phase 1. Keynotes are textual annotations that relate text strings to specific elements in the model, which are in turn linked to an external text file. It can be used as external link to the element itself with specific style and specifications so it can be used as a Revit family. That means, user can insert different text family types in Revit. Keynotes can be assigned to elements which are typically used if the user wants to note an entire assembly, such as a wall assembly. It is very important to parse building components in such data format that is suitable to push into existing database. This data must be consistent as well to suit database structure. Each item that is presented in a separate line in the database to ease and simplify their usage.

Phase 3. It focuses on creating a plug-in, which is a type of algorithm that retrieves preview image for all elements in family files of Revit. Plug-In or Add-In are terms used in BIM tool to signify a module containing an algorithm that makes use of the BIM tool's Application Program Interface (API). The BIM tool used in this study has a .NET API, which means that any of the .NET compliant programming languages

(C#, VB.NET, F#, etc.) can be used to develop a customized plug-in. While each language has its own relative benefits, C# has been used in this research due to its simplicity, usability, and powerful ability to underlay the .NET framework. Code below represents sample of the developed algorithm used to create preview images from each type of family file base on several popular image formats. This algorithm uses C# programming language, which is used in developing the plug-ins that will be applied to the BIM tool.

```
Document doc = uiDoc.Document;
using (Transaction transaction = new Transaction(doc, "Print")) {
    transaction.Start();
    string initialName = GetFileName(doc);
    filePath = SelectFileNameDialog(initialName);
    if (filePath == initialName) return;
    IList<ElementId> views = new List<ElementId>();
    views.Add(doc.ActiveView.Id);
    string normalizedName = new FileInfo(filePath).Name;
    FileInfo imageFile = new FileInfo($"{filePath}{userValues.UserExtension}");
    var tmpFilePath = Path.Combine(imageFile.DirectoryName,
        $"{Guid.NewGuid().ToString()}{imageFile.Extension}");
    FileInfo tmpFile = new FileInfo(tmpFilePath);
    var exportOptions = new ImageExportOptions {
        ExportRange = ExportRange.VisibleRegionOfCurrentView,
        FilePath = tmpFilePath,
        FitDirection = FitDirectionType.Vertical,
        HLRandWFViewsFileType                                              =
GetImageFileType(userValues.UserExtension),
        ImageResolution = userValues.UserImageResolution,
        PixelSize = userValues.UserImageHeight,
        ShouldCreateWebSite = false,
        ShadowViewsFileType = GetImageFileType(userValues.UserExtension),
        ZoomType = ZoomFitType.FitToPage };
    if (views.Count > 0) exportOptions.SetViewsAndSheets(views);
    if (ImageExportOptions.IsValidFileName(filePath))
    doc.ExportImage(exportOptions);
    transaction.Commit(); }
doc.Dispose();
```

Phase 4. It consists of designing REST-service that help first of all retrieving of information from database (in this particular project it is Amazon RDS) in JSON format. This data is being used in main plug-in module, which manipulates it to access particular element from a numerous components of main catalogue. REST-service easily gives an access to elements by category, version etc. This service allows push newly parsed data to the database as well as retrieving data. Of course with the help of authentication such as user’s passport (hashed login and password). This service makes it really convenient to access data throughout the internet even without BIM software such as Revit or AutoCAD. The RDS database is consistent and contains all needed parameters and properties of the elements to speed up catalogue search. Yet, it must be said that the 3D geometric information cannot be stored in database in the following format, therefore database information is not equal to the .rfa family file of Revit.

Phase 5. It concentrates on designing the main module named Revit Family Browser that interconnect the building components information from the database with Revit panel containing preview images. This tool directly loads particular family or type to the project instead of finding it manually through numerous family files with its own types.

Conclusions

Building information modelling and automated quantities technologies can provide the industry with consequential opportunities to raise the quality of the industry to a much higher and sophisticated level. Having the capability to simulate a range of data options with real-time cost advice and carry on throughout the detailed design, construction, and operational stages, BIM will surely place construction practices at a higher value.

Public and private sector clients are increasingly demanding their projects are realized on a BIM platform. Despite some countries adopting it at a faster rate than others, there is consensus that the world’s built-environment will benefit dramatically from the potential inherent in BIM to improve its social and environmental performance. BIM is key in the pressing search for more sustainable solutions and is already moving towards becoming an industry wide approach at a global level.

Because of this, CAD drawings allow you to create drawings and designs in the digital domain that were previously done by hand. The digital format makes data processing easier, safer, and faster. Pre-drawn hand-drawn drawings can be scanned and then expanded digitally. Many CAD programs now use 3D drawings to maximize productivity and deliver faster, higher quality product results by allowing the smallest details to be worked out.

According to the Autodesk Object Model, each Autodesk object has its own properties and methods. Using these methods, you can automate, speed up, and simplify the process of drawing BIM systems through programming.

The models described earlier, and the graphs presented demonstrate the obvious superiority of BIM over any other type of structural design. And automating this process not only simplifies the work of engineers and architects, but also greatly reduces the cost and speeds up development in general, which actively attracts large corporations to use these methods.

PART 2

ANALYZING OF BIM AUTOMATION WITH AMAZON WEB SERVICES AND AUTODESK .NET API

The Autodesk .NET API allows for programmatic management of Autodesk applications and drawing files using submitted collections or libraries. When these objects are opened, they can be accessed in different programming languages and environments [7].

There are several advantages of implementing the .NET API for Autodesk:

- Programmatic access to Autodesk drawings opens up to more programming environments. To the .NET API, developers were limited to ActiveX automation and languages that supported COM, AutoLISP and C ++ using ObjectARX.
- Integration with other Windows-based applications, such as Microsoft Excel and Word, is greatly facilitated by using the program's own .NET API or the open source ActiveX / COM library.
- The .NET Framework is designed for both 32-bit and 64-bit operating systems. Visual Basic for Applications was developed for 32-bit operating systems only.
- Allows access to advanced programming interfaces of a lower learning curve than for traditional programming languages such as C ++.

Objects are the basic building blocks of the Autodesk .NET API. Each public object represents an exact part of Autodesk, and they are grouped into different assemblies and namespaces. There are many different types of objects in the Autodesk .NET API. For example:

- graphic objects, such as lines, arcs, text, and dimensions;
- style settings, such as text and dimensional styles;
- organisational structures, such as layers, groups and blocks;
- drawing displays, such as views and viewing area;
- drawings and Autodesk application.

2.1. Components of the Autodesk .NET API

The Autodesk .NET API is made up of different DLL files that contain a wide range of classes, structures, methods, and events that provide access to objects in a drawing file or the Autodesk application. Each DLL file defines different namespaces which are used to organize the components of the libraries based on functionality [9].

The three main DLL files of the Autodesk .NET API that you will frequently use are:

- **AcDbMgd.dll.** Use when working with objects in a drawing file.
- **AcMgd.dll.** Use when working with the Autodesk application.
- **AcCui.dll.** Use when working with customization files.

Use an Autodesk .NET API DLL

Before you can use classes, structures, methods and events found in one of the dlls associated with the Autodesk .NET API, you must send the dll to the project. After referencing the dll to your project, you can use the namespaces and components in the DLL file in your project.

After referencing the Autodesk .NET API dll, you must set the Copy Local dll reference property to False. The Copy Local property determines whether Microsoft Visual Studio creates a copy of the dll file being sent and places it in the same directory as the project build file when it is built. Since send files are already shipped with Autodesk, creating copies of send dll files can cause unexpected results when the build file is loaded into Autodesk [7].

2.2. Which Edition of Microsoft Visual Studio to Use

Microsoft Visual Studio is available in multiple versions and editions. To use the .NET API for Autodesk, you need to use as minimal:

- Microsoft Visual Studio 2008 with Service Pack 1
- Microsoft .NET Framework 3.5 with Service Pack 1

If you are using AutoCAD 2007 through AutoCAD 2009, you should use:

- Microsoft Visual Studio 2005
- Microsoft .NET Framework 2.0 or later

Microsoft Visual Studio is offered in two versions: free and paid. The free release is known as Microsoft Visual Studio Express Edition, while the addressed versions change in name and cost through the different improvement apparatuses remembered for them. Microsoft Visual Studio Standard Edition is a section level rendition, which gives the improved usefulness of Microsoft Visual Studio Express Edition alongside various different highlights. The most well-known release of Microsoft Visual Studio utilized by designers is Microsoft Visual Studio Professional Edition [7].

In spite of the fact that you can utilize Microsoft Visual Studio Express with the AutoCAD .NET API, this instructional exercise expects you are utilizing one of different versions, for example, Microsoft Visual Studio Standard Edition or Microsoft Visual Studio Professional Edition.

There are four primary points of interest to utilizing Microsoft Visual Studio:

- Robust and open improvement climate that has an unassuming expectation to absorb information.
- VBA and VB.NET linguistic structure are comparable, which makes it an ideal climate for existing VBA clients.
- Visually natural and broad discourse box creation apparatuses.
- Projects can be worked as an independent executable or DLL get together which would then be able to be stacked into AutoCAD for execution.

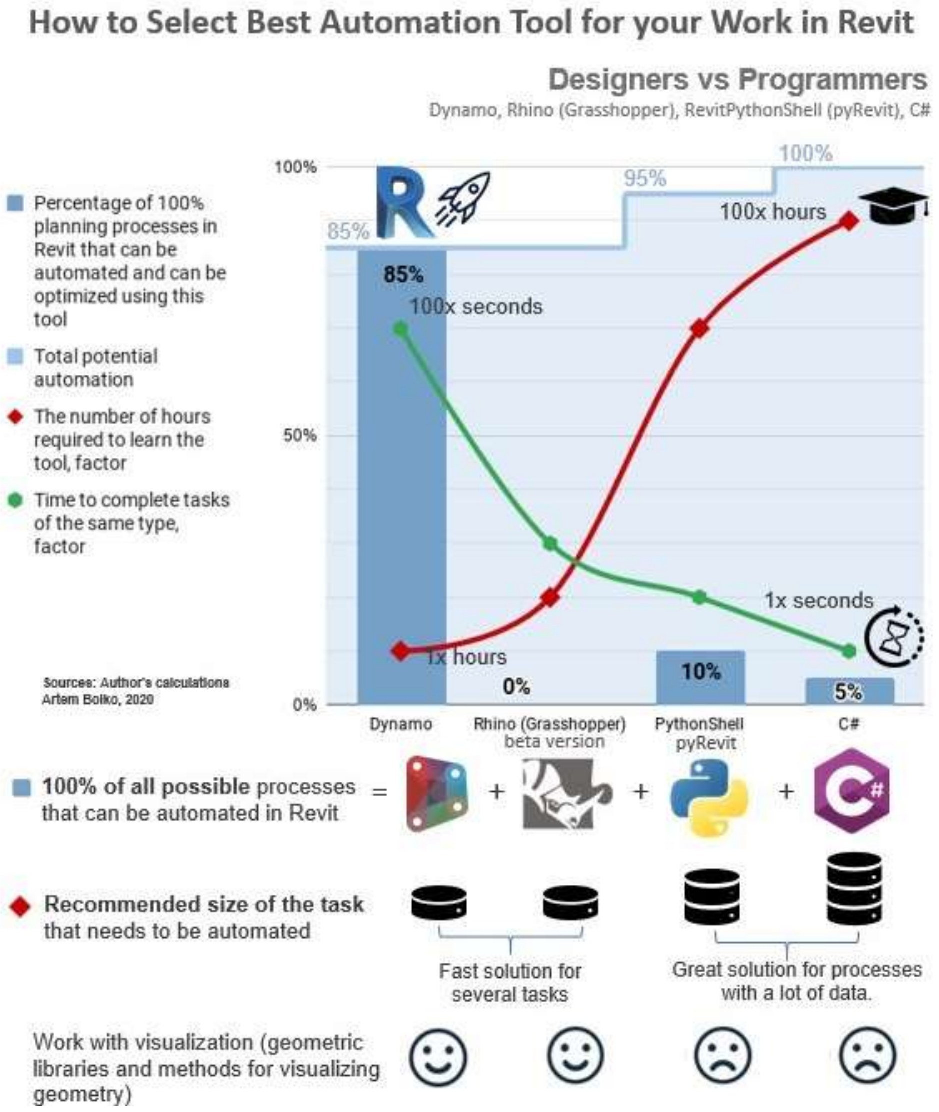
2.3 Revit Design Automation

Design Automation API for Revit can create cloud-native applications that help in building, extracting, and modifying the Revit data. It enables to automate several tasks like documentation, analyzing the model data, creating customized Revit families, and automated report from RVTs [7].

Revit has been around for more than fifteen years. It is a mature product. Everyone noticed, the rate of new feature development has slowed considerably. If you want Revit to do something specific, to make it work the way you work, you are most likely going to need to do it yourself.

Learning to automate Revit is a bit like working on your car or fixing up your house. You are going to look under the hood, get your hands dirty, and learn how things work. You already know how to use the software. Now it is time to take things to the next level and really get productive.

At the moment it is available 4 options for automating work in Autodesk Revit (fig. 2.1-2.2):



Sources: Author's calculations Artem Boiko, 2020

Fig. 2.1. Revit automation tools overview

Fortunately, it is not too hard to get started. There are three options for automating Revit. You can visually program Revit using Dynamo. You can write custom add-ins with Visual Studio. Or, my favorite, you can write macros using Revit's built-in macro editor.

Which tool will save engineers thousands of hours by automating BIM processes in Revit?

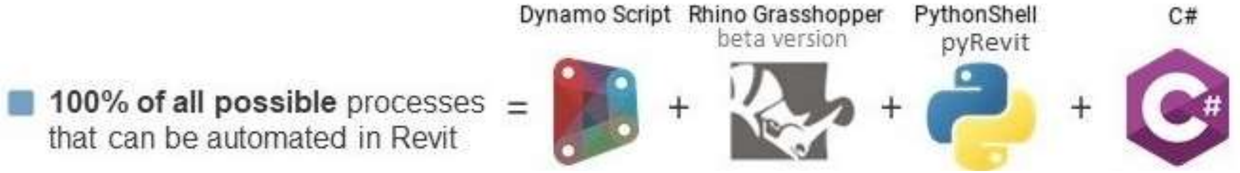


Fig. 2.2. Automating options for Autodesk Revit

- Dynamo (open source graphical programming)
- Rhino.Inside (beta version, with Grasshopper it's like Dynamo)
- PythonShell or pyRevit (Python with Revit API)
- C# (C# with Revit API)

2.3.1 An Overview of the Tools

- Dynamo (fig.2.3) is a visual programming tool that works with Revit. Dynamo extends the capabilities of Revit by making the Revit API (Application Programming Interface) more accessible. Instead of typing code, in Dynamo you create programs by manipulating graphical elements called "nodes." This programming approach is better suited for visually oriented types such as architects, designers, and engineers.

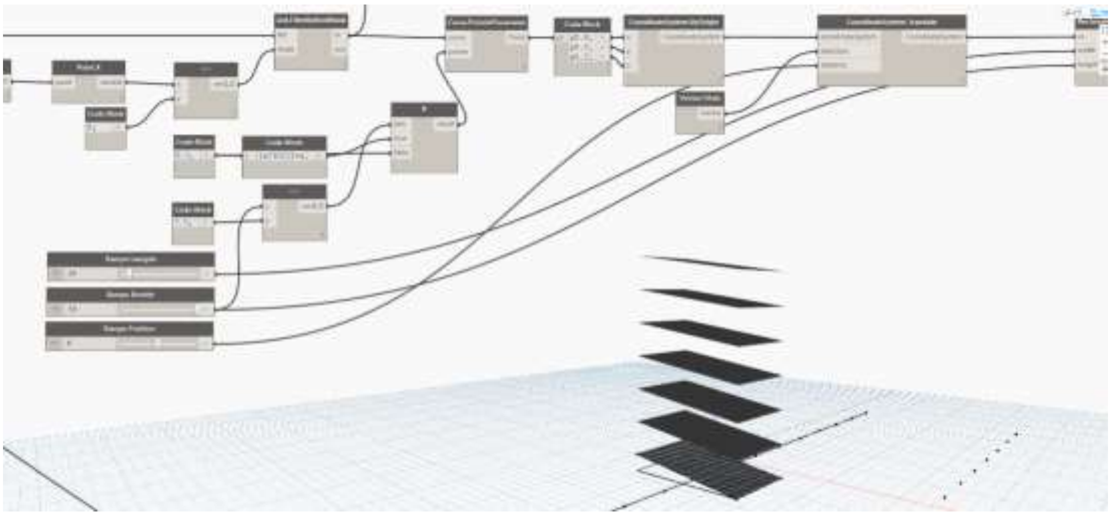


Fig. 2.3. Dynamo visual programming tool

- Rhino.Inside (fig. 2.4) is an open source Rhino WIP project that allows Rhino and Grasshopper to run in other 64-bit Windows applications like Revit, AutoCAD, etc. With Rhino Inside Revit, you also get all the functionality of Rhino. including

Grasshopper for Revit. Grasshopper is a visual programming language similar in concept to Dynamo (although there are fundamental differences).

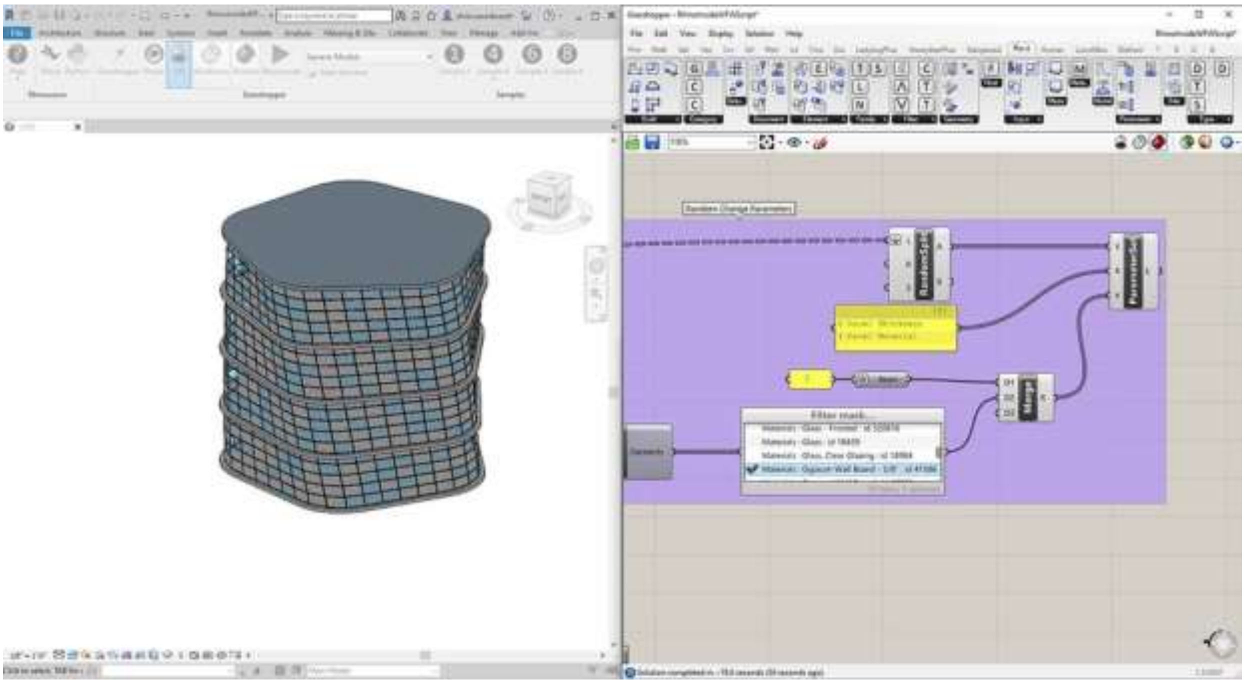
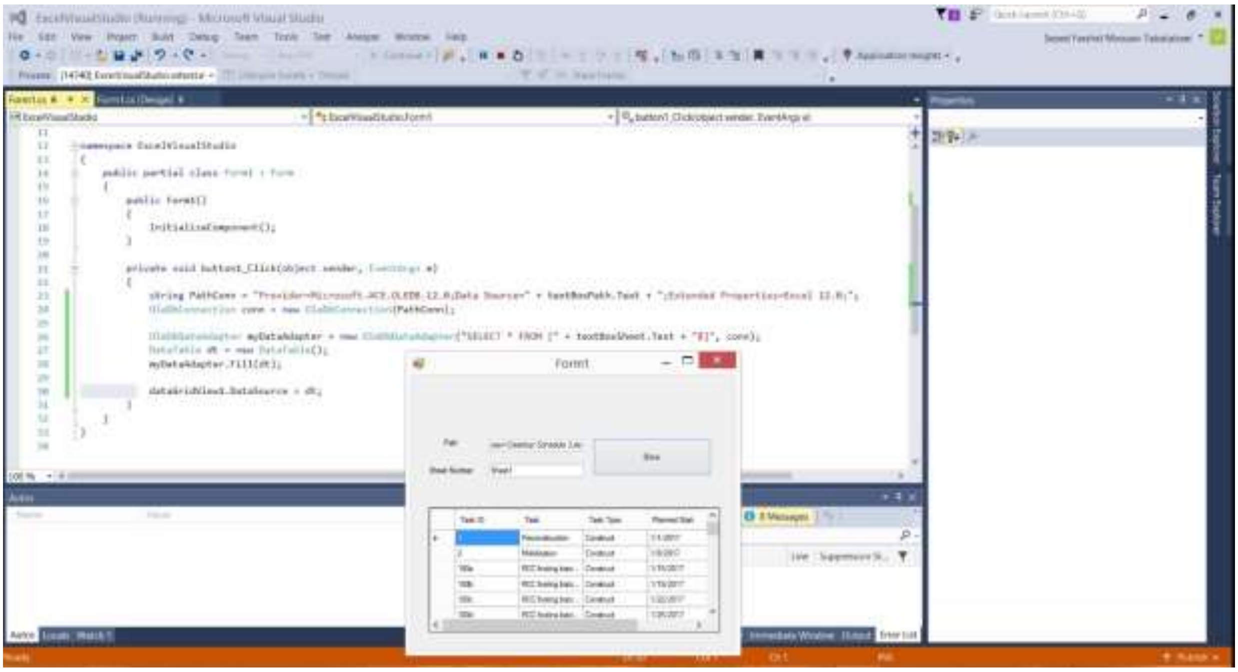

Filter mask...
Materials : Glass - Frosted : id 320016
Materials : Glass : id 18459
Materials : Glass, Clear Glazing : id 18964
Materials : Gypsum Wall Board : 1/8" : id 47106

Fig. 2.4. Rhino.Inside project screenshot

- C# – Autodesk Revit provides a rich API that you can use to customize existing product features or add entirely new ones. You can automate repetitive, time-consuming tasks and expand core functionality without leaving the Autodesk Revit environment. API (fig.2.5) can be used to create custom tools and features that connect directly to Autodesk Revit, extending its functionality.


ExcelVisualStudio (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Form1.cs Form1.cs [Design]
ExcelVisualStudio
ExcelVisualStudio.Form1
button1_Click(object sender, EventArgs e)
Form1
Path
Sheet Number
Sheet1
Autos
Name
Value
Properties
Solution Explorer
Team Explorer

```
namespace ExcelVisualStudio
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            string PathConn = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + textBoxPath.Text + ";Extended Properties=Excel 12.0;";
            OleDbConnection conn = new OleDbConnection(PathConn);

            OleDbDataAdapter myDataAdapter = new OleDbDataAdapter("SELECT * FROM [" + textBoxSheet.Text + "$]", conn);
            DataTable dt = new DataTable();
            myDataAdapter.Fill(dt);

            dataGridView1.DataSource = dt;
        }
    }
}
```

Fig. 2.5. Example of RevitAPI using with simple UI

- RevitPythonShell – Provides scripting capabilities for Autodesk Revit and Project Vasari software. Designers have the ability to interactively design and manipulate Revit elements using algorithm and computational logic with Python code.

2.4. Analyzing Revit API

API is an application programming interface, a medium through which a developer can interconnect with existing software. Revit API is a collection of programs and libraries that helps in building and integrating software to Revit to extend its functionality. It helps to integrate Revit with external programs like RISA and ETABS. The Revit Platform API is compatible with the .NET framework that allows developing applications in any .NET compliant languages like C#, C++, and VB.NET. Revit's .NET API enhances the software functionality in Simulation, Conceptual design, Building management, and construction activities. It allows us to link up with programs external to Revit such as RISA, ETABS, etc [10].

2.4.1 Basic Functionalities and Features of Revit API

The Revit API for Structural elements and workflow analysis is extended with structural code checking, enabling BIM experts and developers to connect several tools like structural analysis calculation, creating, modifying, and numbering of reinforcement, etc. Electrical analysis and the workflow can be embedded in BIM using Revit API. Revit API's are continuously extended for building energy performance analysis, MEP analysis, and Structural and design options.

In Revit, using the Revit API and C#, customized plug-ins can be created. Custom Revit API tools are usually created to enhance productivity and build professional add-ins.

Add-ins are the most powerful method for automating Revit but they are also the most difficult for beginners. Add-ins are installed on the local computer and are accessible from the Revit Add-ins ribbon or a custom Ribbon. Add-ins are written in code (either C# or VB.Net) using an external programming environment like Microsoft Visual Studio. Once you write the add-in, you compile the code as a .dll file and load it into Revit using XML manifest file.

Since add-ins are installed like software, they’re easier to distribute in an office environment. You just need to install the .dll and the XML file in the proper location and every user of that computer can access the add-in. In addition, you can fully customize the way the add-in appears on the ribbon.

One of the difficulties with creating add-ins is the learning curve. It is necessary to learn the Revit API. It is needed to learn how to use Visual Studio. It is necessary to learn how to connect Visual Studio to Revit. Additionally, it is necessary to learn how to create controls in the Revit interface. It takes some time to get comfortable with each of these. That said, add-ins give you the greatest control over Revit and provide the most seamless interface with the software. Plus, creating really useful add-ins, it is possible to sell them on the Autodesk App Store.

Geometry Extraction - with the help of Revit API, we can extract Geometry from the indexed property Element (fig.2.6); Geometry or Element.get_Geometry in C#. By feeding the option 'class' of this property, the output end can be customized such as detail level, view, non-visible elements to be included and compute references, etc. With the custom Exporter API of Revit 2014, the 3D geometry can be extracted. Here is an example of geometry extracted using C# and C++ [10].

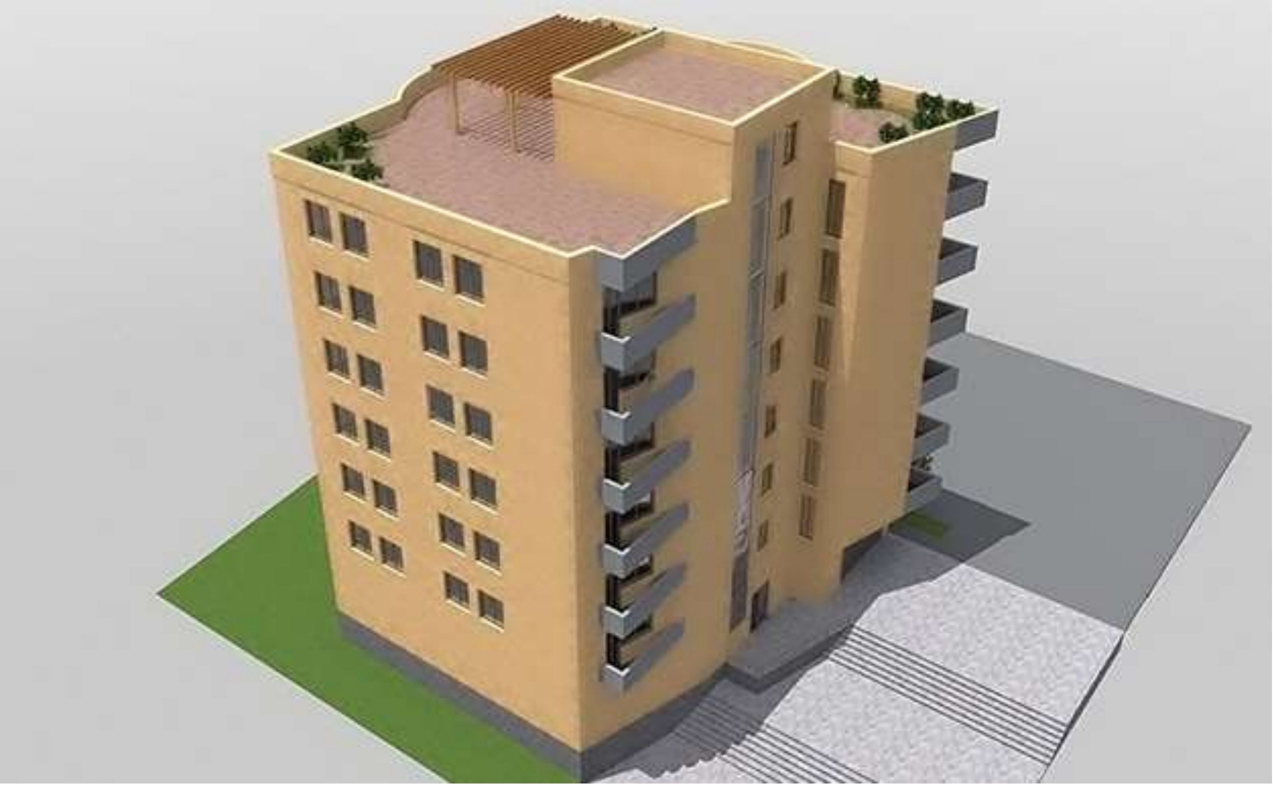


Fig. 2.6. Extracted geometry from Revit via C# with C++

The sectional view of the extracted from the geometry was this (fig.2.7):

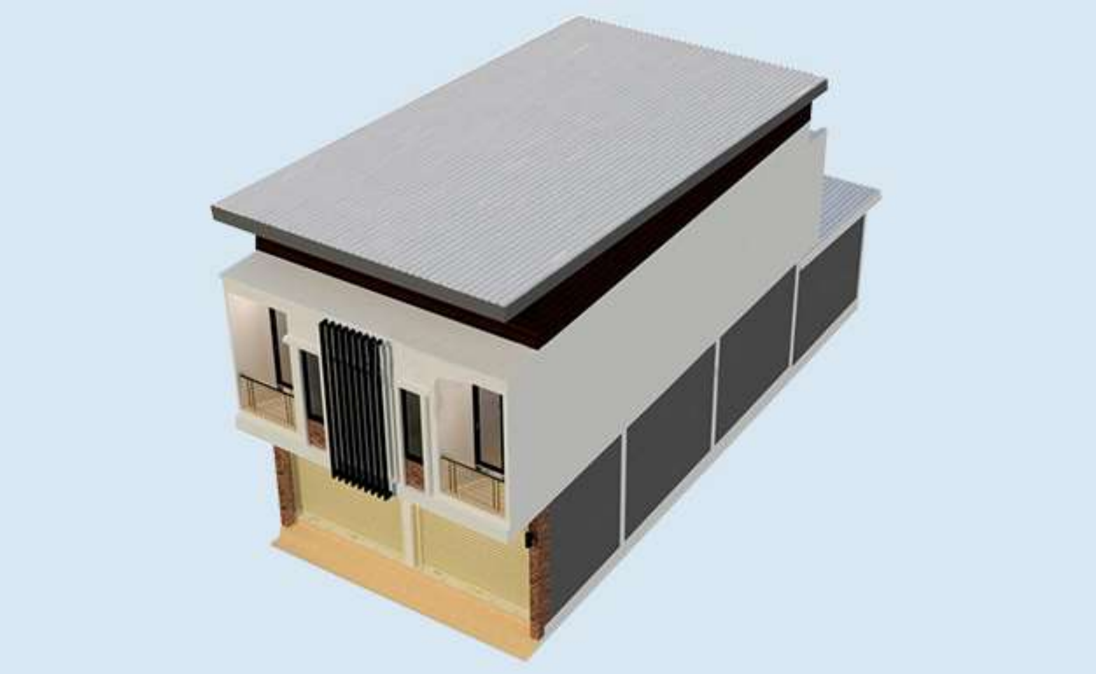


Fig. 2.7. Sectional view of the exctracted geometry

Revit API can be used to extract data from any linked files, RVT File, or Basic File. With the use of Revit API, a parameter data can be exported to Excel and re-imported to the model. For example, a new shared parameter can be created and populated in the model, all the values can be exported to excel and then the modified data re-imported from excel to update the Revit model.

2.4.2 Revit API utilization

There are three main ways how you can utilize Revit API:

- Add-ins
- Macros
- Dynamo

Each of these ways of utilization have advantages and disadvantages. Add-in is the one which require the most time to develop but from other hand it is the easier to use for the end user. Macro gives you a chance to solve more complicated tasks in the fastest way and there is also no limitation in Revit API access compare to Dynamo but the macros are the most difficult to share. Finally, Dynamo gives the most flexibility for the end user who can adjust script without having hard programming skills.

2.4.3 Revit Add-in

The presentation is focused on API utilization via Revit add-in.

There is many great additional tools available on market you can purchase or download for free. Most of these tools are doing great job to automate application of the company standards. However, these add-ins are developed to meet requirements of many workflows and sooner or later this cause that developers need to go on compromise or even abandon automation of some parts of the workflows. This is main reason why company should consider having add-in developed in-house so it will be possible to stream line small and simple tools focused on specific company needs.

2.5. Analyzing Dynamo

Dynamo is a visual programming environment for Revit (fig.2.11). It's open-source software which you can download for free. Of all three methods, Dynamo is the new kid on the block and has generated considerable interest and attention [10].

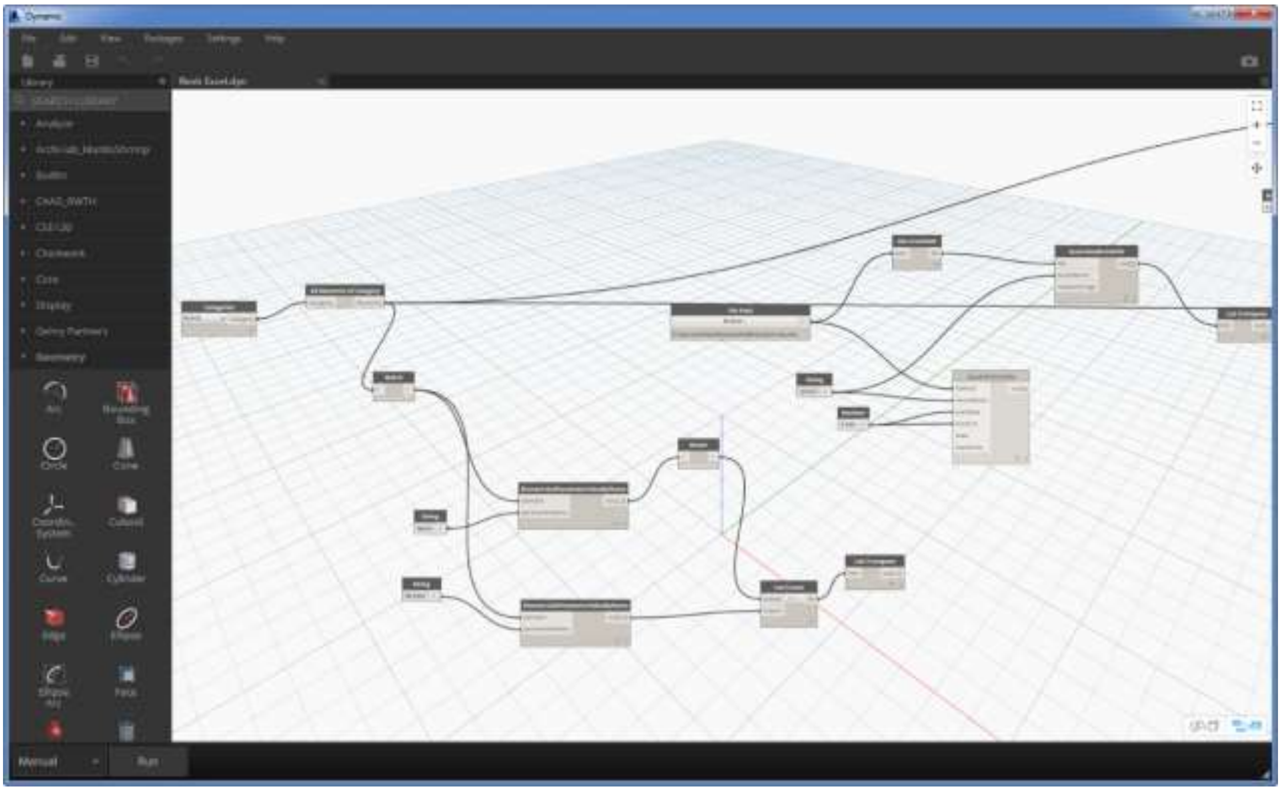


Fig. 2.8. Dynamo application interface screenshot

In Dynamo, you create programs, or "graphs", by connecting code blocks called "nodes". Each node performs a specific function and has inputs and output. The process is very visual. Nodes are connected with lines that determine the sequence of events.

One of the benefits to Dynamo is that you don’t need to know how to program, since the nodes contain all of the programming logic.

On the downside, Dynamo graphs can get really complicated. The more nodes you use, the more connecting lines you have in the graph. After a while, it looks like a lot of spaghetti on the screen. Also, you can’t create an interface for a Dynamo graph, so anyone else who wants to use your graph needs to have Dynamo installed and a working knowledge of the software. Another downside is that Dynamo graphs aren’t compiled so they tend to run slower than the other methods. Lastly, Dynamo is still evolving so it’s not fully mature. . . yet. There are things you can do with add-ins or macros that you can’t do in Dynamo without a little programming know-how.

Dynamo is a visual programming tool that is intuitive to the Revit environment. It creates an algorithm to read, analyze, and edit data. It takes input, then extracts and processes the data to give output, usually in interaction to Revit files.

It is an open-source platform and maintains a single website for learning materials and resources for all needs. The links like, Learn, Forum, and Event help the users to get the learning resource, nodes or packages are shared on forums in addition to queries on forums, and upcoming workshops or conferences are posted on the Event section. Dynamo is an ever-evolving tool and its latest version is launched almost daily. Geometry Library and Design Script are some of the other functionalities provided to users by dynamo [10].

2.5.1 Basic Functionalities of Dynamo

- Visual Programming – this is a visual programming framework where a program is created by using nodes. Here Nodes are boxes having an input and output end, each node (fig.2.9) performs a specific task or function. These nodes are interconnected with wires. The program flow is represented graphically and logically using these nodes.

Some of the nodes of Dynamo interact with Revit API. Dynamo can use Python script nodes for advanced functions which is similar to writing programming software.

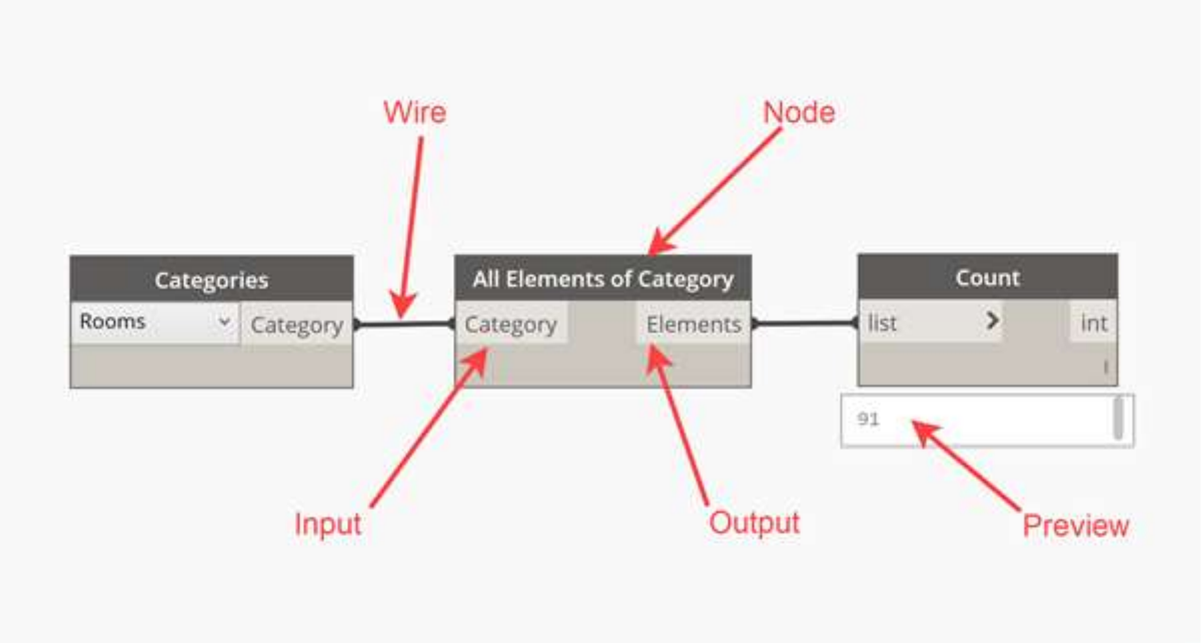


Fig. 2.9. Description of Dynamo nodes and its properties

- Creates its own Geometry with parametric relationship - In parametric design, changes made to one parameter changes the output or results connected with that parameter. Dynamo creates a geometrical relationship with nodes, a hierarchy of geometrical relation is created, where changes made to any node, changes the nodes connected with it. With Dynamo added, these variables can be changed automatically.

- Automates Repetitive tasks - In Revit, we have to perform numerous repetitive tasks. Revit has made it easy to perform repetitive tasks but not like with dynamo. As an example, if we want to create a sheet in Revit, we need to click right first, then left and then need to do a little typing. But in Revit adding 50 sheets is a difficult task, each time we need to click to add a sheet. With Dynamo these tasks can be automated, we can create sheets directly by viewing the models or from an excel file. Using Dynamo, several tasks can be automated by creating tools like flip the grid heads, insert families in batch, renumbering or revision of sheets, changing the text head, etc. [10].

- Generative Design tool - Dynamo is a generative design tool as it create a design using algorithm or a set of rules. The application can be leveraged by the AEC industry to explore and optimize the deign options. Here, designers are allowed to develop geometrical relationships logically irrespective of any traditional sculpting (fig.2.10).

Fig. 2.10. Visualized model of project via Dynamo and Revit

Whether it be geometry for twisting towers, layout of mechanical rooms, layout of furniture in a classroom or glittering patterns on a wall, there are various design applications available with Dynamo.

2.5.2 Limitations of Dynamo Comparing to Revit

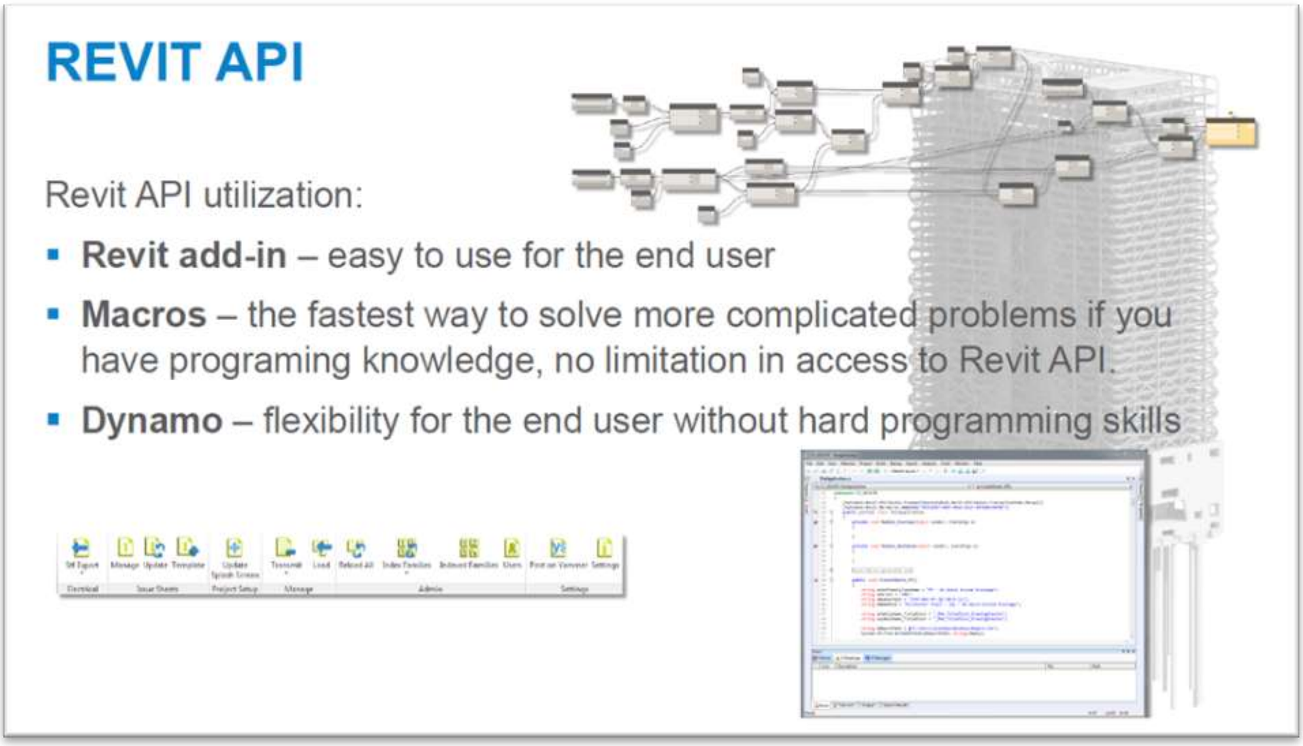
Even though Dynamo is steadily evolving but compared to similar tools, like Grasshopper with Rhino, the application has limitations in designing tools for computational acoustic analysis, structural parametric design, energy analysis, light analysis, etc. The reason may be the platform, i.e., Revit on which Dynamo is built as Revit has some modeling limitations [10].

Dynamo works on graphs and it can get complicated with the increase in the number of nodes, as the number of connecting lines will increase between the nodes.

An interface cannot be created in Dynamo if someone wants to use one’s Dynamo graph. The person needs to install Dynamo to use that graph. Even, Dynamo has been found a bit slower than other methods and has certain limitations in manipulating RVT Files.

2.5.3 Dynamo vs Revit API

Autodesk Revit has been a popular 3D modeling and designing software used by construction companies, architects, and designers in all parts of the world. Dynamo integration to Revit has made it more powerful in terms of automating repetitive tasks, exploring hundreds of design options, performance testing, and many more. At the same time, Dynamo has a specific set of features that can be leveraged at various stages of projects. To find the Revit versus Dynamo information pointers to make the best use of their features and design/modeling capabilities helps image below (fig. 2.11) [10].



REVIT API

Revit API utilization:

- **Revit add-in** – easy to use for the end user
- **Macros** – the fastest way to solve more complicated problems if you have programing knowledge, no limitation in access to Revit API.
- **Dynamo** – flexibility for the end user without hard programming skills

Fig. 2.11. RevitAPI combinations description

Irrespective of these limitations, Dynamo is still one of the better options for architects and the AEC Industry to enhance its project design process and efficiency. The user-friendly and visual programming approach helps the users to do more with less effort and achieve better results.

2.6 Amazon Web Services

Amazon's cloud platform provides almost every feature in the cloud computing industry. Their cloud services allow easy access to computing power, storage or other features needed by application developers. AWS has many products that fall into different categories. In addition to the features mentioned above, they offer developer tools, management tools, mobile services and application services. As you understand, application services combined with computing infrastructure and database are critical components for a successful enterprise mobile application development team.

In addition to a wide range of services, AWS Cloud has adjusted its cloud computing prices since its inception in 2006. Their pricing is very competitive with all other cloud providers. Prices for their cloud services have continued to fall due to competition and pricing structures. AWS offers free tiers of service to startups and individuals. It's an easy way to try before you buy. Moreover, development teams can buy servers by the second rather than by the hour. Depending on what services the team uses, you can probably find a reasonable pricing structure for AWS that is lower than the cost of the entire infrastructure investment [12].

In addition, the Amazon Web Services cloud platform offers developers more than 15 years of enterprise infrastructure. Because admin teams like AWS are constantly working to improve the platform, your development team can benefit from their expertise. In terms of management abilities and skills, AWS has some of the best talent on the market. Of course, you will want to choose a platform that has a lot of experience.

2.6.1 AWS Pros and Cons

Amazon's biggest strength is its dominance of the public cloud market. In its Magic Quadrant for Cloud Infrastructure as a Service, Worldwide, Gartner noted, "AWS has been the market share leader in cloud IaaS for over 10 years." Some of the most relevant AWS Services are shown on the Figure 2.12.

Part of the reason for its popularity is undoubtedly the massive scope of its operations. AWS has a huge and growing array of available services, as well as the most comprehensive network of worldwide data centers. The Gartner report summed it up, saying, "AWS is the most mature, enterprise-ready provider, with the deepest capabilities for governing a large number of users and resources."

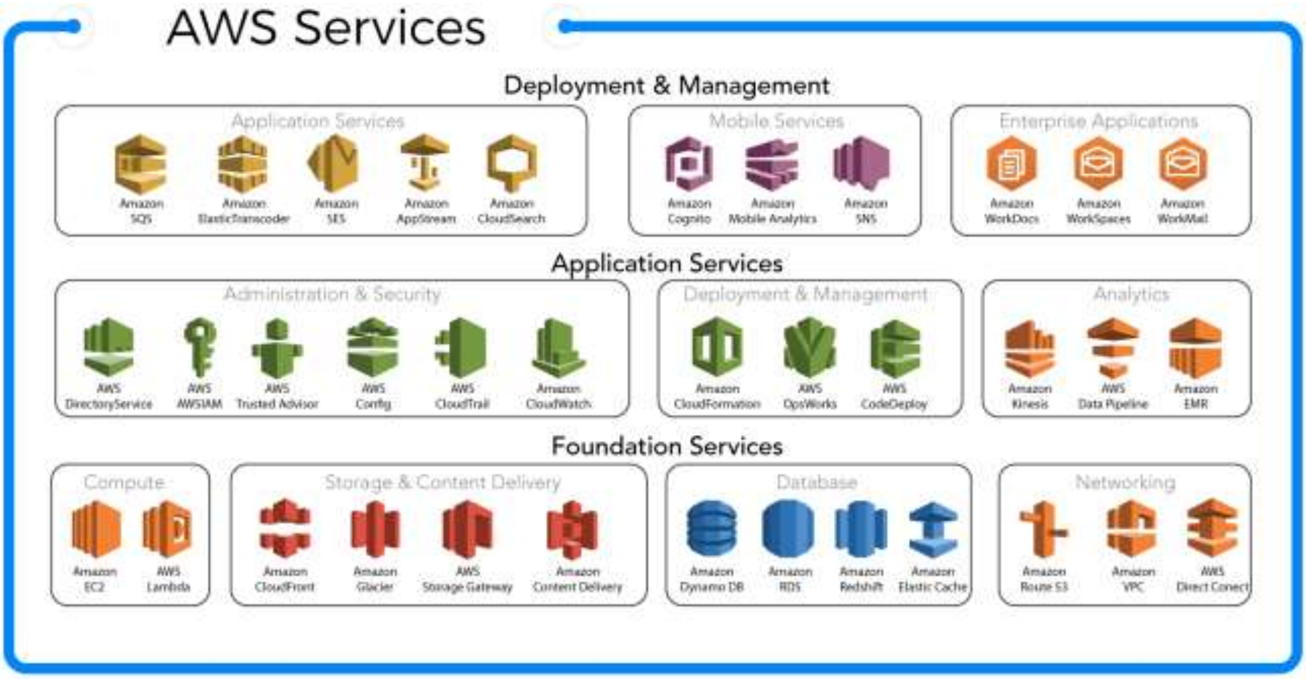


Fig. 2.12. Available AWS Services

Amazon's big weakness relates to cost. While AWS regularly lowers its prices, many enterprises find it difficult to understand the company's cost structure and to manage those costs effectively when running a high volume of workloads on the service [12].

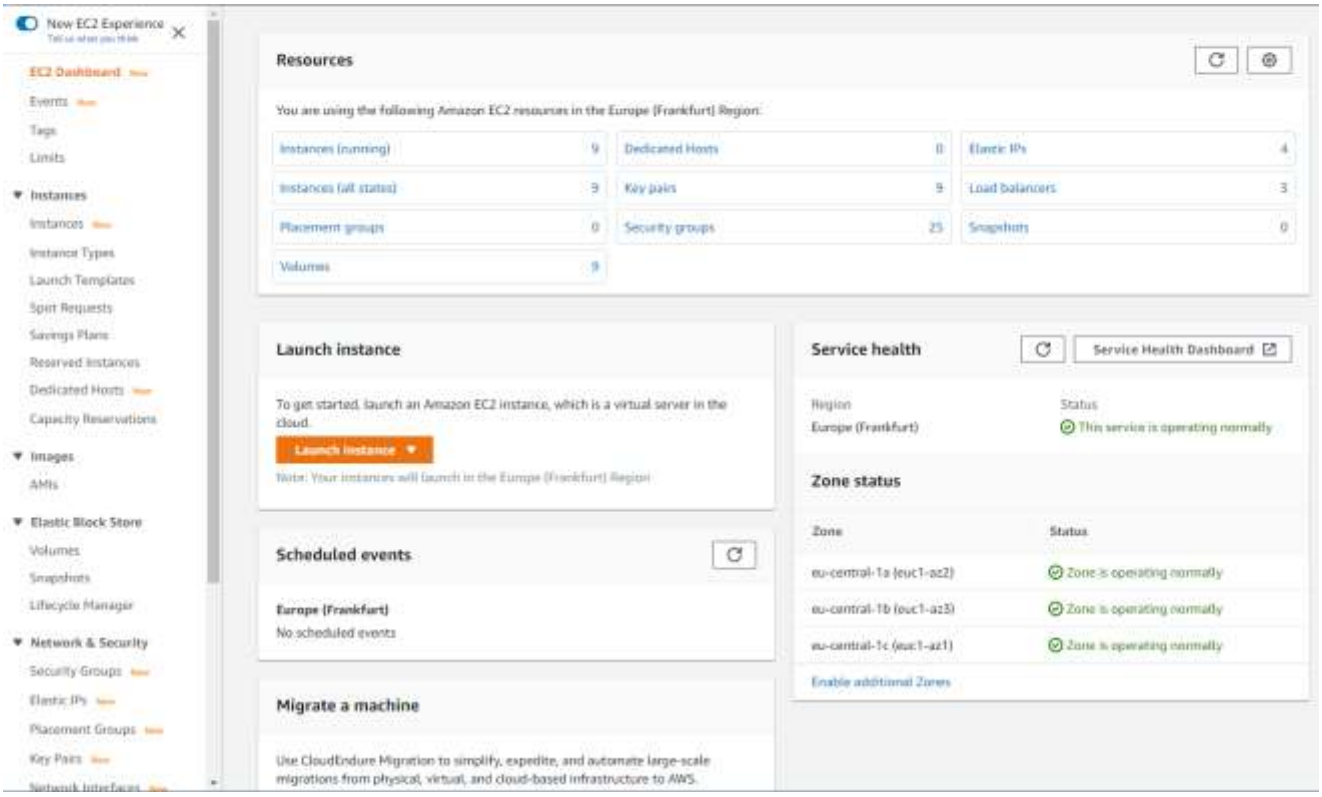
In general, however, these cons are more than outweighed by Amazon's strengths, and organizations of all sizes continue to use AWS for a wide variety of workloads.

2.7. AWS Features Overview

2.7.1 Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud (fig.2.13). It is designed to make web-scale cloud computing easier for developers. Amazon EC2’s simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon’s proven computing environment [13].

Amazon EC2 offers the broadest and deepest computing platform with a choice of processor, storage, network, operating system and acquisition model (Figure 2.14). We offer the fastest processors in the cloud and we are the only cloud with a 100Gbps Ethernet network. We have powerful GPU instances for machine learning and graphics workloads, as well as the instances with the lowest price per output in the cloud. There are more SAP, HPC, machine learning and Windows workloads running on AWS than any other cloud [13].



Resources

You are using the following Amazon EC2 resources in the Europe (Frankfurt) Region:

Instances (running)	9	Dedicated Hosts	0	Elastic IPs	4
Instances (all states)	9	Key pairs	9	Load balancers	3
Placement groups	0	Security groups	25	Snapshots	0
Volumes	9				

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the Europe (Frankfurt) Region

Scheduled events

Europe (Frankfurt)
No scheduled events

Migrate a machine

Use CloudEndure Migration to simplify, expedite, and automate large-scale migrations from physical, virtual, and cloud-based infrastructure to AWS.

Service health

Region	Status
Europe (Frankfurt)	This service is operating normally

Zone status

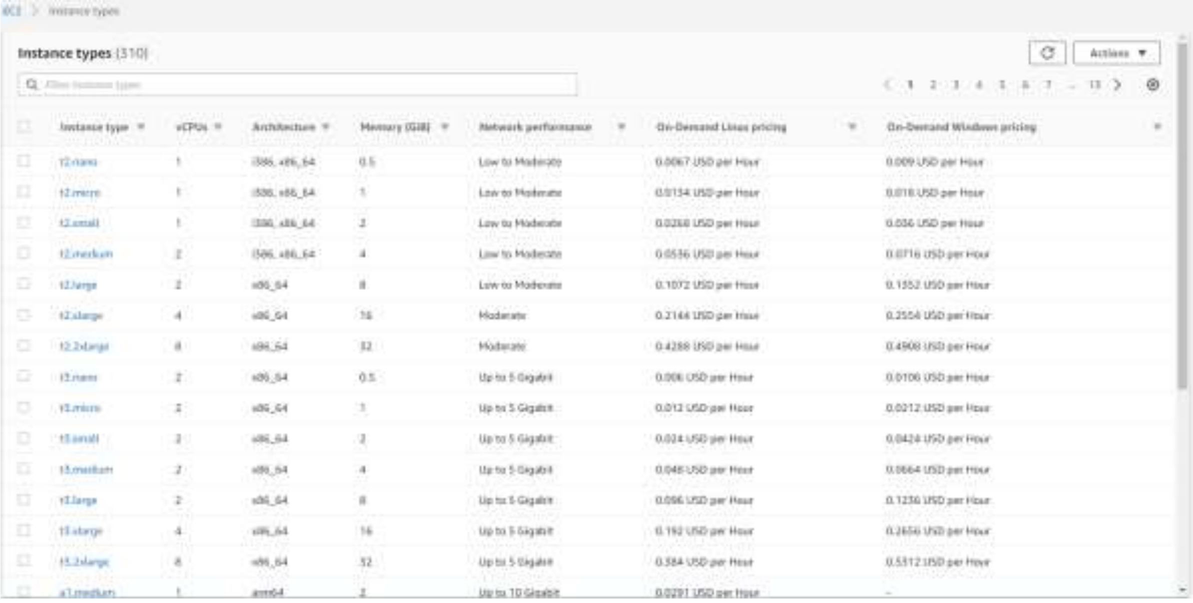
Zone	Status
eu-central-1a (euc1-az2)	Zone is operating normally
eu-central-1b (euc1-az3)	Zone is operating normally
eu-central-1c (euc1-az1)	Zone is operating normally

Fig. 2.13. Amazon EC2 dashboard screenshot with existing instances

Amazon EC2 offers a wide and deep selection of instances built on the latest compute, storage and networking technologies and is designed for high performance and security.

- Faster innovation and enhanced security with AWS Nitro - The AWS Nitro system is the core platform for our next generation of EC2 instances, offloading many traditional virtualization features into dedicated hardware and software to deliver high performance, high availability and high security. while reducing virtualization overhead. The Nitro system is a rich collection of building blocks that can be assembled in a variety of ways, giving us the flexibility to design and rapidly deliver new types of EC2 instances with an ever-expanding choice of compute, storage, memory and networking options.

- Choice of processors - the choice of the latest generation Intel Xeon, AMD EPYC and AWS Graviton processors provides the best balance of performance and price for your workloads. EC2 instances powered by NVIDIA GPUs and AWS Inferentia are also available for workloads requiring accelerated computing, such as machine learning, games and graphics programs.



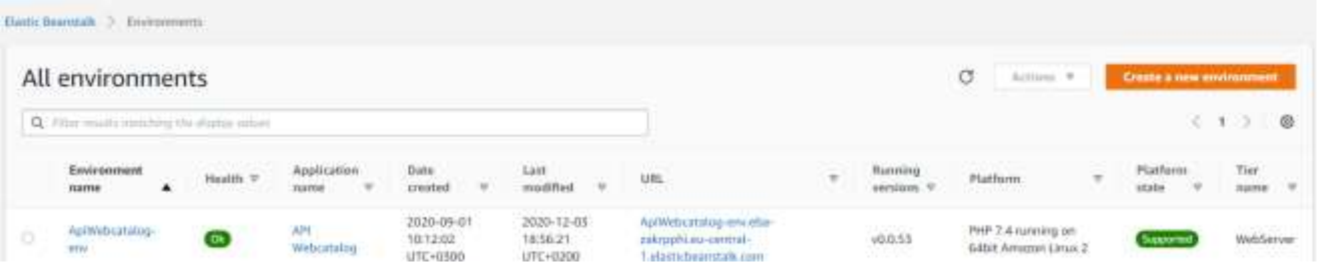
EC2 > Instance types

Instance types (310)

□	Instance type ▼	vCPUs ▼	Architecture ▼	Memory (GiB) ▼	Network performance ▼	On-Demand Linux pricing ▼	On-Demand Windows pricing ▼
□	t2.nano	1	i386, x86_64	0.5	Low to Moderate	0.0067 USD per Hour	0.009 USD per Hour
□	t2.micro	1	i386, x86_64	1	Low to Moderate	0.0134 USD per Hour	0.018 USD per Hour
□	t2.small	1	i386, x86_64	2	Low to Moderate	0.0268 USD per Hour	0.036 USD per Hour
□	t2.medium	2	i386, x86_64	4	Low to Moderate	0.0536 USD per Hour	0.0716 USD per Hour
□	t2.large	2	x86_64	8	Low to Moderate	0.1072 USD per Hour	0.1552 USD per Hour
□	t2.xlarge	4	x86_64	16	Moderate	0.2144 USD per Hour	0.2554 USD per Hour
□	t2.2xlarge	8	x86_64	32	Moderate	0.4288 USD per Hour	0.4908 USD per Hour
□	t3.nano	2	x86_64	0.5	Up to 5 Gigabit	0.006 USD per Hour	0.0106 USD per Hour
□	t3.micro	2	x86_64	1	Up to 5 Gigabit	0.012 USD per Hour	0.0212 USD per Hour
□	t3.small	2	x86_64	2	Up to 5 Gigabit	0.024 USD per Hour	0.0424 USD per Hour
□	t3.medium	2	x86_64	4	Up to 5 Gigabit	0.048 USD per Hour	0.0664 USD per Hour
□	t3.large	2	x86_64	8	Up to 5 Gigabit	0.096 USD per Hour	0.1236 USD per Hour
□	t3.xlarge	4	x86_64	16	Up to 5 Gigabit	0.192 USD per Hour	0.2656 USD per Hour
□	t3.2xlarge	8	x86_64	32	Up to 5 Gigabit	0.384 USD per Hour	0.5312 USD per Hour
□	a1.medium	1	arm64	2	Up to 10 Gigabit	0.0291 USD per Hour	-

Fig. 2.14. Examples of Amazon EC2 instance types

- High performance storage - Amazon Elastic Block Store (EBS) provides easy to use, high performance block storage for use with Amazon EC2 (fig.2.15). Amazon EBS is available in a range of volume types that allow you to optimize storage performance and cost for your workloads. Many EC2 instance types also come with options for local NVMe SSD storage for applications that require low latency [13].



Elastic Beanstalk > Environments

All environments

	Environment name ▲	Health ▼	Application name ▼	Date created ▼	Last modified ▼	URL ▼	Running versions ▼	Platform ▼	Platform state ▼	Tier name ▼
○	ApiWebcatalog-env	Ok	API Webcatalog	2020-09-01 10:12:02 UTC+0300	2020-12-03 18:56:21 UTC+0200	ApiWebcatalog-env.eba-zakrppbi.eu-central-1.elasticbeanstalk.com	v0.0.53	PHP 7.4 running on 64bit Amazon Linux 2	Supported	WebServer

Fig. 2.15. Amazon EBS created application example

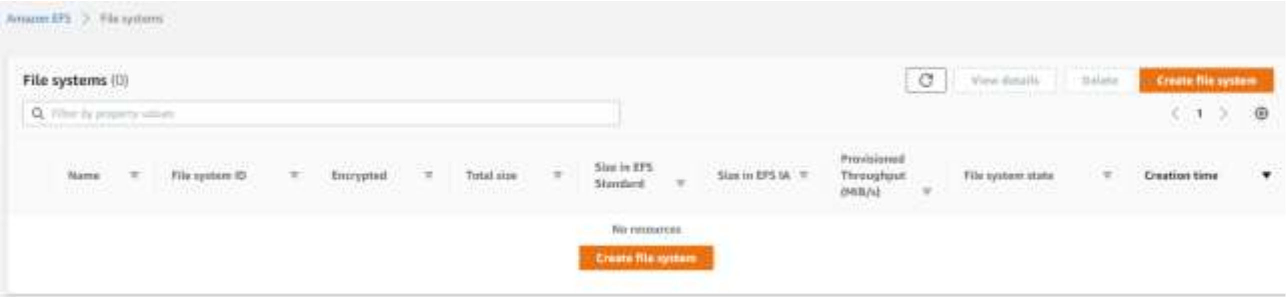
- Enhanced network - AWS is the first and only cloud to offer 100Gbps enhanced Ethernet for compute instances. The enhanced network enables significantly higher packet per second (PPS), less network jitter and lower latency. For high performance computing (HPC), the Elastic Fabric Adapter is a network interface for Amazon EC2 instances that offers low-latency, high-bandwidth interconnectivity between compute nodes to help scale programs to thousands of cores.

- Choice of purchasing model - offers a choice of several purchasing models with request-to-order, spot instances and a savings plan. With Spot Instance, you can save up to 90% on failover workloads. With a savings plan, you can save up to 72% with dedicated usage and flexibility in EC2, Fargate and Lambda. You can also optimise your costs with instance recommendations built into EC2 with AWS Compute Optimizer or with tools such as Explorer [13].

Different Amazon EC2 workloads can have very different storage requirements. In addition to embedded instance storage, we also have Amazon Elastic Block Store (Amazon EBS) and Amazon Elastic File System (Amazon EFS) to meet other cloud storage workload requirements (Figure 2.16).

Amazon EBS provides a persistent, highly available, consistent, then we low latency memory block for use with Amazon EC2 instances. Each Amazon EBS volume automatically replicates within its availability zone to protect you from component failure, ensuring high availability and longevity. It's designed for application managers who need to tune the workload for power, performance and cost.

Amazon EFS provides simple, scalable, persistent, fully managed cloud file storage for shared access. Designed for high availability and longevity across multiple availability zones, it provides a file system interface with standard file system access semantics, automatically increases and decreases capacity, and provides high bandwidth and consistently low latency to application managers at the petabyte scale.



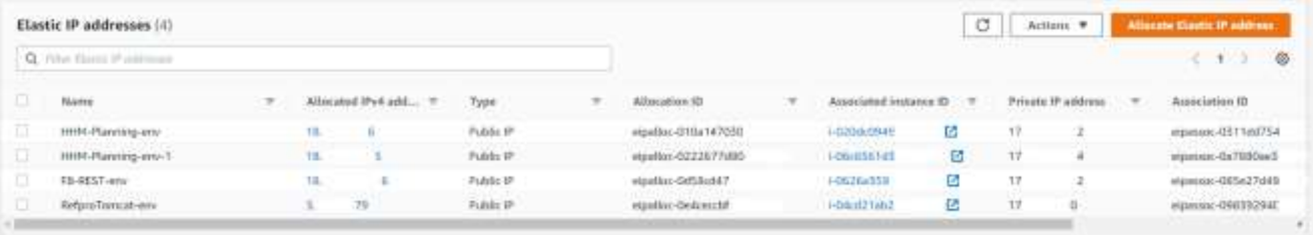
Name	File system ID	Encrypted	Total size	Size in EFS Standard	Size in EFS IA	Provisioned Throughput (MiB/s)	File system state	Creation time

No resources

Fig. 2.16. Amazon EFS creating dashboard

Another important thing to mention is that Amazon EC2 allows you to host instances in multiple locations. Amazon EC2 locations consist of regions and availability zones. Availability zones are individual locations designed to protect against failures in other availability zones and provide low-cost, low-latency network connectivity to other availability zones in the same region. By running instances in separate availability zones, you can protect your programs from the failure of a single location. Regions consist of one or more availability zones, are geographically distributed, and will be located in separate geographical areas or countries. Amazon EC2's service level commitment is 99.99% availability for each Amazon EC2 region. Please refer to Regional Products and Services for more information on the availability of our products and services by region [13].

Elastic IP addresses are static IP addresses designed for dynamic cloud computing (Figure 2.17). An elastic IP address is associated with your account rather than a specific instance, and you manage this address until you decide to explicitly release it. However, unlike traditional static IP addresses, elastic IP addresses allow you to mask instance failure or reachability by programmatically reassigning your public IP address to any instance in your account. Rather than waiting for a data processing technician to reconfigure your host or wait for DNS propagation to all of clients, Amazon EC2 allows to develop problems with your instance or software by quickly reassigning your elastic IP address to the instance replaces. In addition, it is possible to configure a reverse DNS record of any of your elastic IP addresses by filling out this form.



Elastic IP addresses (4)

□	Name	Allocated IPv4 add...	Type	Allocation ID	Associated instance ID	Private IP address	Association ID
□	HHM-Planning-env	18. 6	Public IP	eipalloc-010a147050	i-020dc094E	17 2	eipassoc-0511dd754
□	HHM-Planning-env-1	18. 5	Public IP	eipalloc-0222677d80	i-06d0561d3	17 4	eipassoc-0a7880ae3
□	FB-REST-env	18. 6	Public IP	eipalloc-0d58cd47	i-0626a558	17 2	eipassoc-085e27d49
□	RefproTomcat-env	5. 79	Public IP	eipalloc-0edcecctf	i-0dcd21ab2	17 0	eipassoc-09839294E

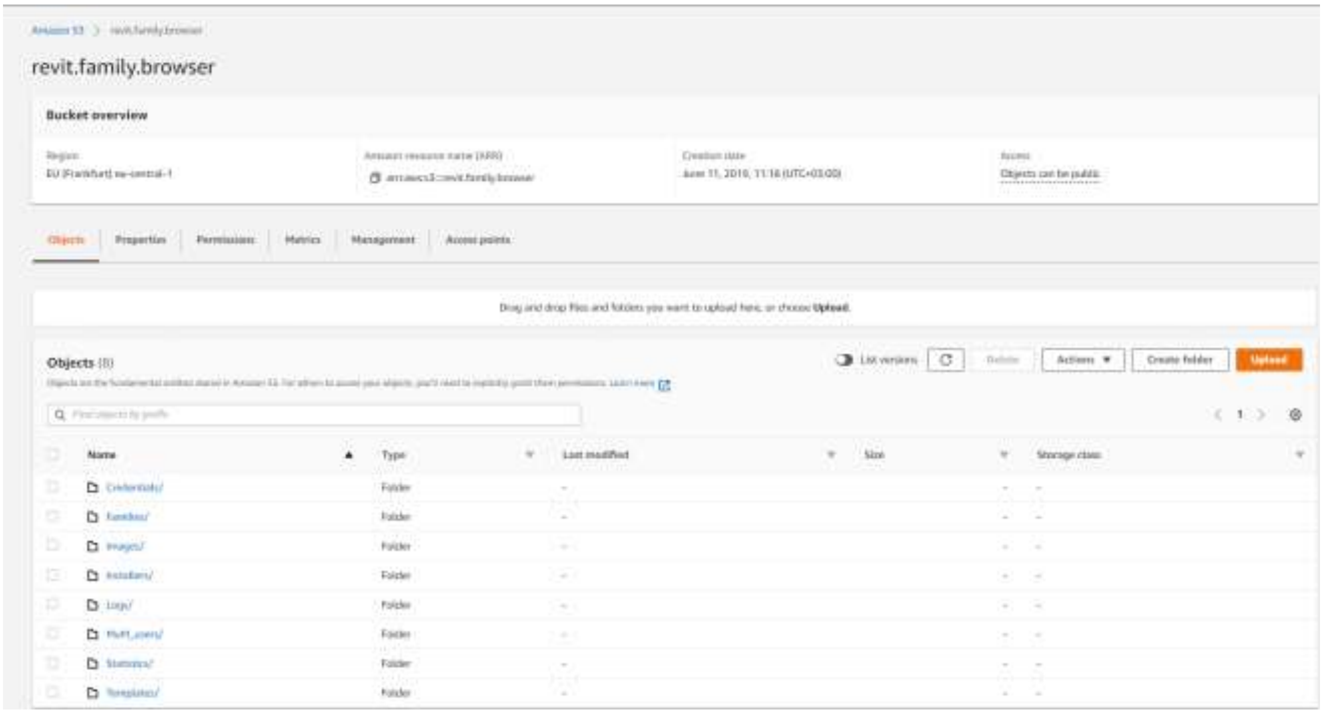
Fig. 2.17. Amazon Elastic IP dashboard

Amazon EC2 Auto Scaling allows you to automatically scale your Amazon EC2 capacity up or down according to conditions you define. With EC2 Auto Scaling, you can ensure that the number of Amazon EC2 instances you’re using scales up seamlessly during demand spikes to maintain performance, and scales down automatically during demand lulls to minimize costs. EC2 Auto Scaling is particularly well suited for applications that experience hourly, daily, or weekly variability in usage. EC2 Auto Scaling is enabled by Amazon CloudWatch and available at no additional charge beyond Amazon CloudWatch fees [13].

Customers can privately access Amazon EC2 APIs from their Amazon Virtual Private Cloud (VPC) or over AWS Direct Connect, without using public IPs, and without requiring the traffic to traverse across the Internet. AWS PrivateLink is a purpose-built technology designed for customers to access Amazon services in a high performance and highly available manner, while keeping all the network traffic within the AWS network. In order to use Amazon EC2 with AWS PrivateLink, you will need to create an endpoint for EC2 in your VPC. Any traffic destined to this endpoint will get privately routed to the EC2 service.

2.8.2. Amazon Simple Storage Service

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance (fig.2.18). This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics [8]. Amazon S3 provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements. Amazon S3 is designed for 99.999% of durability, and stores data for millions of applications for companies all around the world [14].



revit.family.browser

Bucket overview

Region
EU (Frankfurt) eu-central-1

Amazon resource name (ARN)
arn:aws:s3:::revit.family.browser

Creation date
June 11, 2019, 11:16 (UTC+03:00)

Access
Objects can be public

Drag and drop files and folders you want to upload here, or choose **Upload**.

Objects (8)

Objects are the fundamental entities stored in Amazon S3. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
Credentials/	Folder	-	-	-
Families/	Folder	-	-	-
Images/	Folder	-	-	-
Installers/	Folder	-	-	-
Logs/	Folder	-	-	-
Mvrt_users/	Folder	-	-	-
Statistics/	Folder	-	-	-
Templates/	Folder	-	-	-

Fig. 2.18. Amazon S3 bucket screenshot of existing FamilyBrowser application

Industry-leading performance, scalability, availability and durability are the options S3 provides. Easily scale any storage resource up and down to meet fluctuating needs, without upfront investment or resource procurement cycles. Amazon S3 is designed for +99.999% data longevity because it automatically creates and stores copies of all S3 objects across multiple systems. This means your data is available when needed and protected from failures, errors and threats.

There is a wide range of economic classes of storage. Save costs without compromising performance by storing data between S3 storage classes that support different levels of data access at the appropriate speed. You can use S3 storage class analysis to identify data that should move to a lower storage class based on access patterns and configure an S3 lifecycle policy to perform the transfer. You can also store data with changed or unknown access patterns in S3 Intelligent-Tiering, which aggregates objects based on access pattern changes and automatically provides cost savings. With S3 Outposts storage class, you can meet data retention requirements and store data locally in your Outposts environment using S3 on Outposts [14].

S3 provides robust access control, cost, replication and data protection capabilities (Figure 2.19). S3 Access Points make it easy to manage access to data with specific permissions for your applications using a common data set. S3 Replication manages data replication within a region or to other regions. S3 batch operations help manage large-scale changes to billions of objects. Because S3 works with AWS Lambda, you can log actions, identify alerts and automate workflows without managing additional infrastructure.

Run big data analytics on S3 objects (and other data sets in AWS) with Amazon onsite query services. Use Amazon Athena to query S3 data using standard SQL expressions and Amazon Redshift Spectrum to analyze the data that's stored in your AWS data stores and S3 resources. It is possible also to use S3 Select to retrieve subsets of object data rather than the entire object, and improve query efficiency by up to 400%.

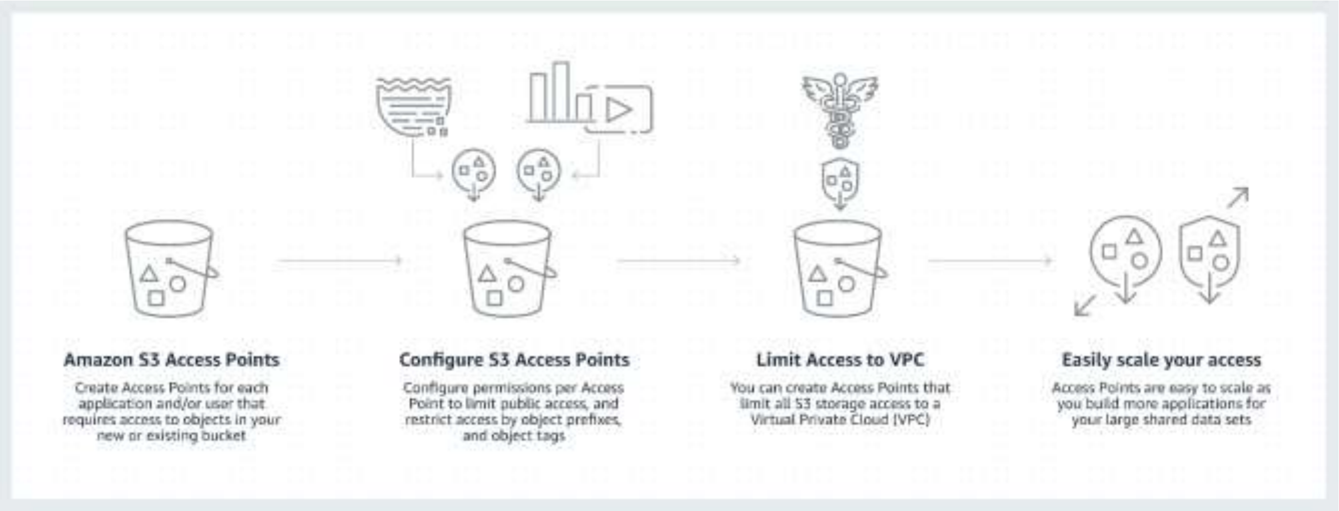

Amazon S3 Access Points
Create Access Points for each application and/or user that requires access to objects in your new or existing bucket
Configure S3 Access Points
Configure permissions per Access Point to limit public access, and restrict access by object prefixes, and object tags
Limit Access to VPC
You can create Access Points that limit all S3 storage access to a Virtual Private Cloud (VPC)
Easily scale your access
Access Points are easy to scale as you build more applications for your large shared data sets

Fig. 2.19. How it works — S3 Access Points

Amazon S3 access points make it easier to manage data access at scale for programs using shared datasets on S3. With S3 access points, you can now easily create hundreds of access points per segment, introducing a new way to provide access to shared datasets. Access points provide an individual path to a segment with a unique host name and an access policy ensuring that specific permissions and network controls are enforced for any request made through the access point [14].

Create a scalable, durable and secure backup and recovery solution using Amazon S3 and other AWS services such as S3 Glacier, Amazon EFS and Amazon EBS to augment or replace existing local capabilities. AWS partners and APNs can help with this through recovery time objectives (RTOs), recovery point objectives (RPOs) and compliance requirements. With AWS, you can back up data you already have in the AWS cloud, or use AWS Storage Gateway, a hybrid storage service, to send backups of local data to AWS.

Protect critical data, applications, and IT systems that run in the AWS cloud or in your on-premises environment without the expense of a second physical site. With Amazon S3 storage, S3 cross-regional replication, and other AWS compute, network, and database services, you can build DR architectures to recover from failures caused by natural disasters, system failures, and human error quickly and easily.

Accelerate innovation by building a data lake on Amazon S3 and gain valuable insights through onsite queries, analytics and machine learning tools. As the data lake grows, use S3 access points to easily configure data access with specific permissions for each programme or set of programmes. You can also use AWS Lake Formation to quickly create a data lake and centrally define and enforce security, management and auditing policies. The service collects data in your databases and S3 resources, moves it to a new data lake in Amazon S3, and cleans and classifies it using machine learning algorithms. All AWS resources are scalable to your extended data stores - with no upfront investment [14].

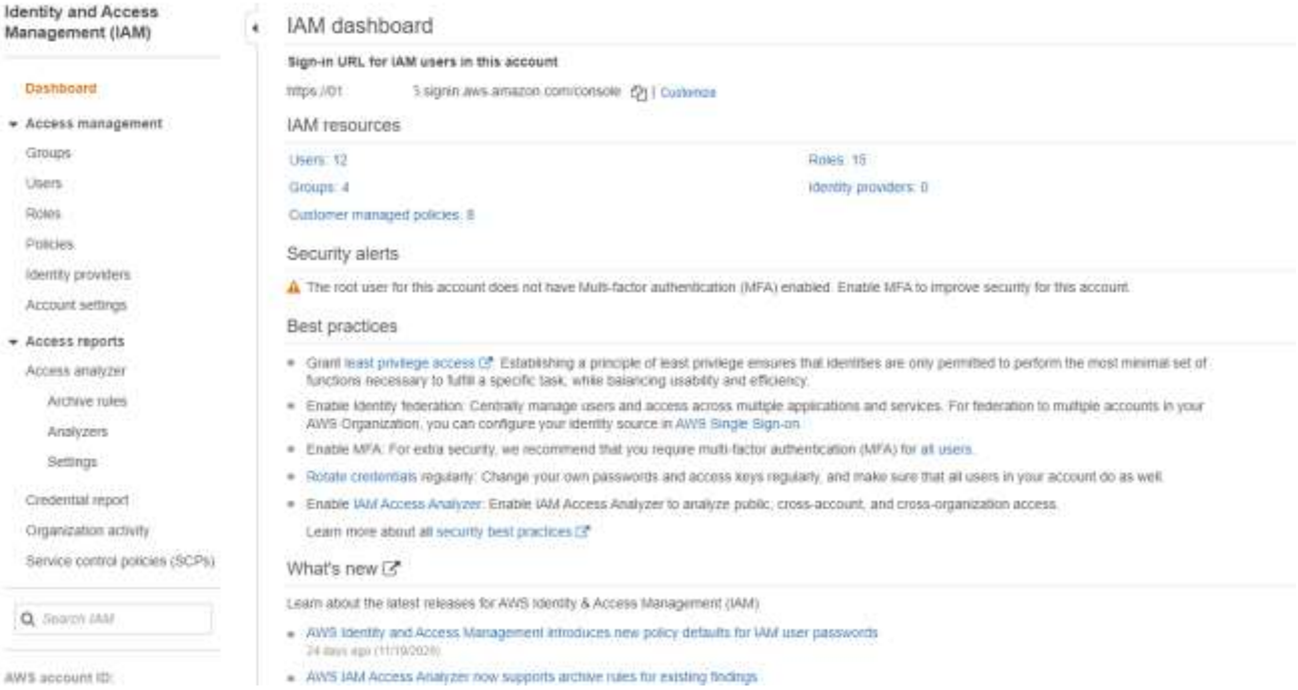
With Amazon S3, any amount of data can be uploaded and accessed anywhere to deploy applications faster and reach more end users. Storing data in Amazon S3 means you see the latest AWS developer tools, S3 APIs and machine learning and analytics services to innovate and optimise your own cloud applications.

2.7.3. AWS Identity and Access Management

AWS Identity and Access Management enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.

IAM is a feature of your AWS account offered at no additional charge (fig.2.20). You will be charged only for use of other AWS services by your users. IAM enables your users to control access to AWS service APIs and to specific resources. IAM also enables you to add specific conditions such as time of day to control how a user can use AWS, their originating IP address, whether they are using SSL, or whether they have authenticated with a multi-factor authentication device [15].

Protect any AWS environment using AWS MFA, a security feature that is available at no additional cost, increases username and password credentials. MFA requires users to confirm physical possession of an MFA hardware token or MFA-enabled mobile device by providing a valid MFA code.



IAM dashboard

Sign-in URL for IAM users in this account

https://01 5.signin.aws.amazon.com/console | Customize

IAM resources

Users: 12 Roles: 15

Groups: 4 Identity providers: 0

Customer managed policies: 8

Security alerts

⚠ The root user for this account does not have Multi-factor authentication (MFA) enabled. Enable MFA to improve security for this account.

Best practices

- Grant least privilege access. Establishing a principle of least privilege ensures that identities are only permitted to perform the most minimal set of functions necessary to fulfill a specific task, while balancing usability and efficiency.
- Enable identity federation: Centrally manage users and access across multiple applications and services. For federation to multiple accounts in your AWS Organization, you can configure your identity source in AWS Single Sign-on.
- Enable MFA: For extra security, we recommend that you require multi-factor authentication (MFA) for all users.
- Rotate credentials regularly: Change your own passwords and access keys regularly, and make sure that all users in your account do as well.
- Enable IAM Access Analyzer: Enable IAM Access Analyzer to analyze public, cross-account, and cross-organization access.

Learn more about all security best practices

What's new

Learn about the latest releases for AWS Identity & Access Management (IAM)

- AWS Identity and Access Management introduces new policy defaults for IAM user passwords
  24 days ago (11/19/2020)
- AWS IAM Access Analyzer now supports archive rules for existing findings

Fig. 2.20. Amazon IAM service dashboard

IAM helps analyse access in your AWS environment. Your security teams and administrators can quickly confirm that your policies provide only implied access to your resources for public and shared accounts. You can also easily define and refine your policy to allow access only to services that are in use. This helps you better adhere to the principle of least privilege [15].

IAM can be used to grant your employees and applications federated access to the AWS Management Console and AWS service APIs, using your existing identity systems such as Microsoft Active Directory. You can use any identity management solution that supports SAML 2.0, or feel free to use one of our federation samples (AWS Console SSO or API federation).

AWS IAM allows to:

- Manage IAM users and their access – You can create users in IAM, assign them individual security credentials (in other words, access keys, passwords, and multi-factor authentication devices), or request temporary security credentials to provide users access to AWS services and resources. You can manage permissions in order to control which operations a user can perform.

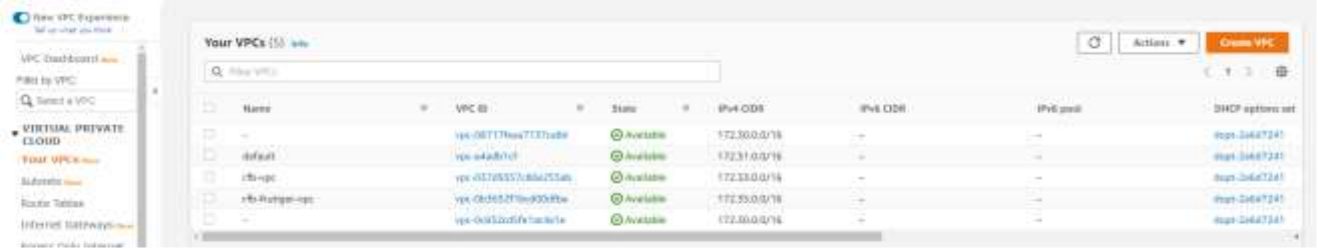
- Manage IAM roles and their permissions – You can create roles in IAM and manage permissions to control which operations can be performed by the entity, or AWS service, that assumes the role. You can also define which entity is allowed to assume the role. In addition, you can use service-linked roles to delegate permissions to AWS services that create and manage AWS resources on your behalf.
- Manage federated users and their permissions – You can enable identity federation to allow existing identities (users, groups, and roles) in your enterprise to access the AWS Management Console, call AWS APIs, and access resources, without the need to create an IAM user for each identity.

2.7.4 Amazon Virtual Private Cloud

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications [16].

It is easy to customize the network configuration of any Amazon VPC. For example, you can create a public-facing subnet for your web servers that have access to the internet. You can also place your backend systems, such as databases or application servers, in a private-facing subnet with no internet access. You can use multiple layers of security, including security groups and network access control lists, to help control access to Amazon EC2 instances in each subnet.

Amazon VPC provides advanced security features, such as security groups and network access control lists, to enable inbound and outbound filtering at the instance and subnet level (fig.2.21). In addition, you can store data in Amazon S3 and restrict access so that it’s only accessible from instances inside your VPC. For additional security, you can create dedicated instances that are physically isolated from other AWS accounts, at the hardware level.



Your VPCs (5) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	IPv6 pool	DHCP options set
–	[illegible]	Available	172.30.0.0/16	–	–	[illegible]
default	[illegible]	Available	172.31.0.0/16	–	–	[illegible]
cfb-vpc	[illegible]	Available	172.33.0.0/16	–	–	[illegible]
cfb-[illegible]-vpc	[illegible]	Available	172.35.0.0/16	–	–	[illegible]
–	[illegible]	Available	172.30.0.0/16	–	–	[illegible]

Fig. 2.21. Amazon Virtual Private Cloud management console

Creation of VPC is quick and easy using the AWS Management Console. Select from common network setups and find the best match for your needs. Subnets, IP ranges, route tables, and security groups are automatically created. You spend less time setting up and managing, so you can concentrate on building the applications that run in your VPCs [16].

Manage your virtual network environment, including selecting your own range of IP addresses, creating subnets and setting up route tables and network gateways. Set up your network configuration by, for example, creating a public subnet for your web servers that has access to the Internet, and placing your internal systems, such as databases or application servers, on a private subnet without access to the Internet.

Host a basic web application such as a blog or simple website in a VPC, and get the added layers of privacy and security that Amazon VPC provides. You can help secure your site by creating security group rules that allow web servers to respond to incoming HTTP and SSL requests from the Internet while preventing web servers from initiating outgoing connections to the Internet. You can create a VPC that supports this use case by selecting "VPC with only one public subnet" in the Amazon VPC console wizard [16].

Host tiered web applications and strictly enforce access and security restrictions between your web servers, application servers, and databases. Run web servers on a public subnet while running application servers and databases on private subnets so that application servers and databases have no direct access from the Internet. You control access between servers and subnets through inbound and outbound packet filtering, network access control lists, and security groups. To create a VPC that supports this use case, you can select "VPC with public and private subnets" in the Amazon VPC console wizard.

Using Amazon VPC for disaster recovery, you can get all the benefits of a disaster recovery site for a fraction of the cost. You can periodically backup important data from your data center to a small number of Amazon EC2 instances using Amazon Elastic Block Store (EBS) volumes or import virtual machine images into Amazon EC2. To ensure business continuity, you can run replacement computing capacity in AWS. When the disaster is over, you can send your critical data back to your data center and terminate the Amazon EC2 instances you no longer need. An IPsec VPN connection between Amazon VPC and corporate network encrypts all communication between the application servers in the cloud and databases in your data center. Web servers and application servers in your VPC can leverage Amazon EC2 elasticity and Auto Scaling features to grow and shrink as needed. You can create a VPC to support this use case by selecting "VPC with Public and Private Subnets and Hardware VPN Access" in the Amazon VPC console wizard [16].

Amazon VPC traffic mirroring duplicates the traffic, along with full payload data, from elastic network interfaces (ENIs) of EC2 instances, and delivers it to out-of-band monitoring and security analysis tools.

Amazon VPC ingress routing allows you to easily deploy network and security appliances, including third-party offerings, inline to the inbound or outbound Amazon VPC traffic. Inline traffic inspection helps you screen and secure traffic to protect your workloads from malicious actors.

Conclusions

During this part there was exhibited with various models that Revit is constructed basically for BIM. Both Revit and BIM are significant for mankind and our future in itself. Revit helps originators, structures, various architects to configuration, reproduce, envision, and team up to benefit from the upsides of the interconnected information inside a BIM model. Learn Revit now and gain trust in controlling the information with a BIM model and perceive how transforms you make to one item in a model naturally reflects all through the plan. Getting settled with how plan information is associated will give an unmistakable edge over understudy companions and future collaborators that are more familiar with working with 2D drawings.

One more of the benefits of BIM is the expanding number of reproduction devices that permit fashioners to picture such things as the daylight during various seasons or to measure the computation of building energy execution. The insight of the product, especially Autodesk Revit, to apply decides that depend on material science and best practices gives a supplement to engineers and other task colleagues. The product can do substantially more of the examination and displaying to accomplish top execution, consolidating information and decides into a help that can run with the snap of a catch. In current displaying frameworks the capacity to work with programming frameworks is set up at the center level. Designers just need to pick programming improvement framework associated with API to make ventures.

An integrated development environment Visual Studio 2017 and C# programming language was chosen for the creation of the system because of the exhaustive documentation, flexibility of this framework and Revit API, which support only C#, VB and Python.

Amazon Web Services is a huge platform which provides all the best and new cloud technologies for any need and for any price range. First things first, Amazon enjoys the first-mover advantage. Amazon ventured into cloud computing with AWS as early as 2006 when everyone was skeptical about the potential cloud technology could have. Jeff Bezos did not have the assurance of market research and forecasts. He turned an idea into a booming reality. Bezos, by his own admission, has acknowledged the unusual advantage of a seven-year head start before some competition arrived resulting into far superior and functionally evolved services

AWS is a leader of cloud platforms and a giant with such amount of options, settings and support level. It gives all the most vital benefits for application. This cloud infrastructure vendor offers a wide range of products and solutions delivered over the Internet. With AWS services, it is easy to build practically whatever enterprise application you want in a cost-effective way. At the same time, you can be sure that AWS-based solution is secure, scalable, and reliable.

PART 3

CREATION OF INFRASTRUCTURE WITH PLUG-INS FOR ENGINEERING EQUIPMENT MODELING AND CLOUD TECHNOLOGIES

3.1. Create plug-in for Autodesk Revit

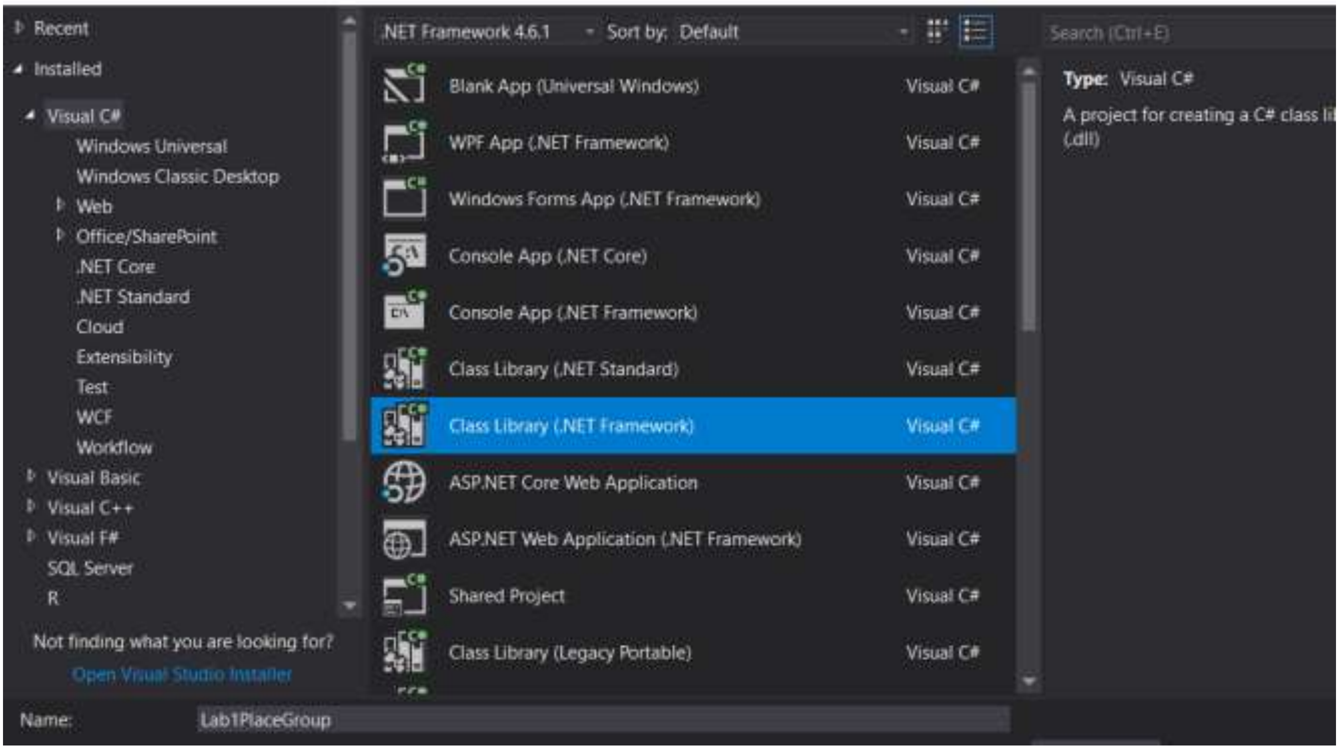
Steps to make Revit plug-in:

1. Launch the Visual Studio development environment:

2. Open Visual Studio (VS) using the Windows Start menu, selecting All Programs, and then Microsoft Visual Studio or any other suitable method to open it.

3. Create a class library project. For this inside VS, in the various files options, click "New Project". In the Installed Templates tab in the left window, find Visual Studio Community and click it. In the main window, select Class Library (fig. 3.1).

4. Enter any name of the future project in the Name box. Then click OK [17].


Recent
Installed
Visual C#
Windows Universal
Windows Classic Desktop
Web
Office/SharePoint
.NET Core
.NET Standard
Cloud
Extensibility
Test
WCF
Workflow
Visual Basic
Visual C++
Visual F#
SQL Server
R
Not finding what you are looking for?
Open Visual Studio Installer
.NET Framework 4.6.1
Sort by: Default
Search (Ctrl+E)
Blank App (Universal Windows) Visual C#
WPF App (.NET Framework) Visual C#
Windows Forms App (.NET Framework) Visual C#
Console App (.NET Core) Visual C#
Console App (.NET Framework) Visual C#
Class Library (.NET Standard) Visual C#
Class Library (.NET Framework) Visual C#
ASP.NET Core Web Application Visual C#
ASP.NET Web Application (.NET Framework) Visual C#
Shared Project Visual C#
Class Library (Legacy Portable) Visual C#

Type: Visual C#

A project for creating a C# class li (.dll)

Name: Lab1PlaceGroup

Fig. 3.1. Create a class library project

VS will create a default template with blank code project for you and display the code in the editor view.

5. Add necessary RevitAPI references: in the Solution Explorer window on the right side of the VS window, right-click References and click Add Reference… (fig. 3.2.)

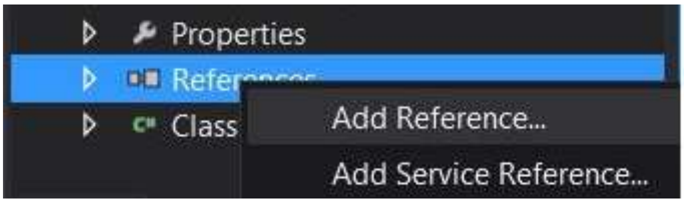
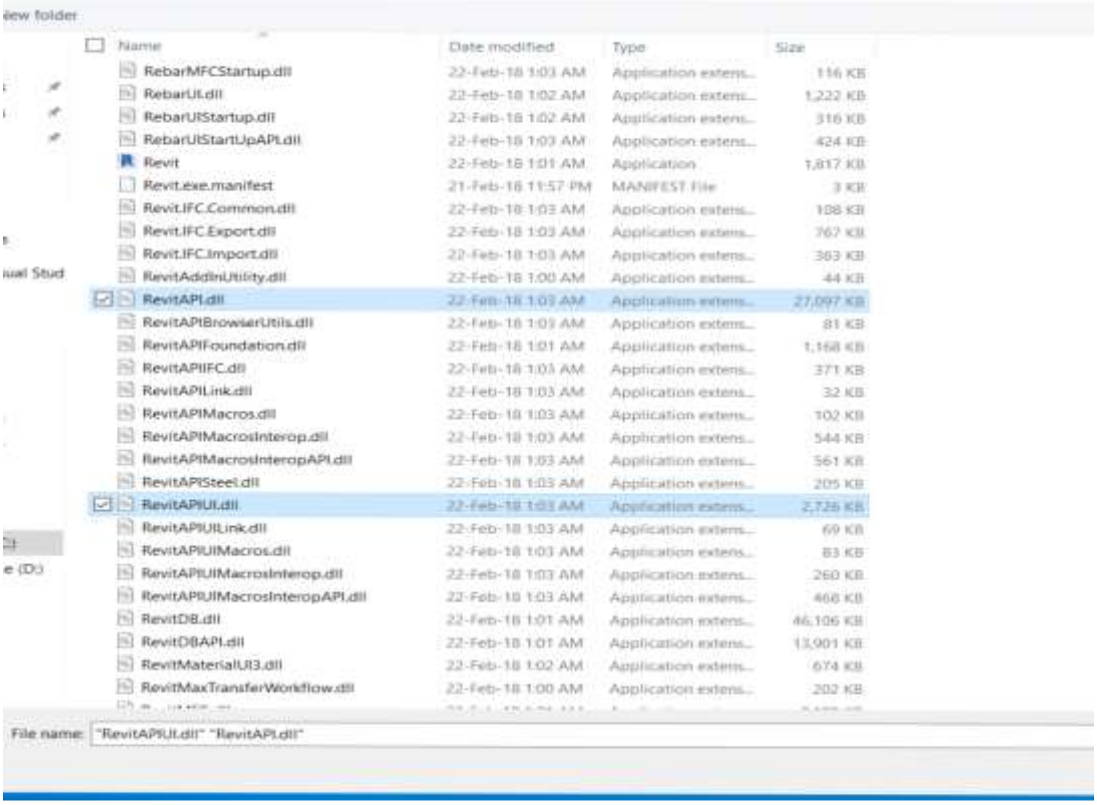


Fig 3.2. Sub-window Add Reference

6. Click the Browse tab and in the Add Reference dialog and browse to the Revit product installation sub-folder. The sub-folder path depends on where you have installed Revit 20xx. The default path is C:\Program Files\Autodesk\Revit 20xx* (fig. 3.3).



□	Name	Date modified	Type	Size
	RebarMFCStartup.dll	22-Feb-18 1:03 AM	Application extens...	116 KB
	RebarUI.dll	22-Feb-18 1:02 AM	Application extens...	1,222 KB
	RebarUIStartup.dll	22-Feb-18 1:02 AM	Application extens...	316 KB
	RebarUIStartUpAPI.dll	22-Feb-18 1:03 AM	Application extens...	424 KB
	Revit	22-Feb-18 1:01 AM	Application	1,817 KB
	Revit.exe.manifest	21-Feb-18 11:57 PM	MANIFEST File	3 KB
	Revit.IFC.Common.dll	22-Feb-18 1:03 AM	Application extens...	108 KB
	Revit.IFC.Export.dll	22-Feb-18 1:03 AM	Application extens...	767 KB
	Revit.IFC.Import.dll	22-Feb-18 1:03 AM	Application extens...	363 KB
	RevitAddInUtility.dll	22-Feb-18 1:00 AM	Application extens...	44 KB
☑	RevitAPI.dll	22-Feb-18 1:03 AM	Application extens...	27,097 KB
	RevitAPIBrowserUtils.dll	22-Feb-18 1:03 AM	Application extens...	81 KB
	RevitAPIFoundation.dll	22-Feb-18 1:01 AM	Application extens...	1,168 KB
	RevitAPIIFC.dll	22-Feb-18 1:03 AM	Application extens...	371 KB
	RevitAPILink.dll	22-Feb-18 1:03 AM	Application extens...	32 KB
	RevitAPIMacros.dll	22-Feb-18 1:03 AM	Application extens...	102 KB
	RevitAPIMacrosInterop.dll	22-Feb-18 1:03 AM	Application extens...	544 KB
	RevitAPIMacrosInteropAPI.dll	22-Feb-18 1:03 AM	Application extens...	561 KB
	RevitAPISteel.dll	22-Feb-18 1:03 AM	Application extens...	205 KB
☑	RevitAPIUI.dll	22-Feb-18 1:03 AM	Application extens...	2,726 KB
	RevitAPIUILink.dll	22-Feb-18 1:03 AM	Application extens...	69 KB
	RevitAPIUIMacros.dll	22-Feb-18 1:03 AM	Application extens...	83 KB
	RevitAPIUIMacrosInterop.dll	22-Feb-18 1:03 AM	Application extens...	260 KB
	RevitAPIUIMacrosInteropAPI.dll	22-Feb-18 1:03 AM	Application extens...	468 KB
	RevitDB.dll	22-Feb-18 1:01 AM	Application extens...	46,106 KB
	RevitDBAPI.dll	22-Feb-18 1:01 AM	Application extens...	13,901 KB
	RevitMaterialUI3.dll	22-Feb-18 1:02 AM	Application extens...	674 KB
	RevitMaxTransferWorkflow.dll	22-Feb-18 1:00 AM	Application extens...	202 KB

File name: "RevitAPIUI.dll" "RevitAPI.dll"

Fig. 3.3. Revit product installation sub-folder

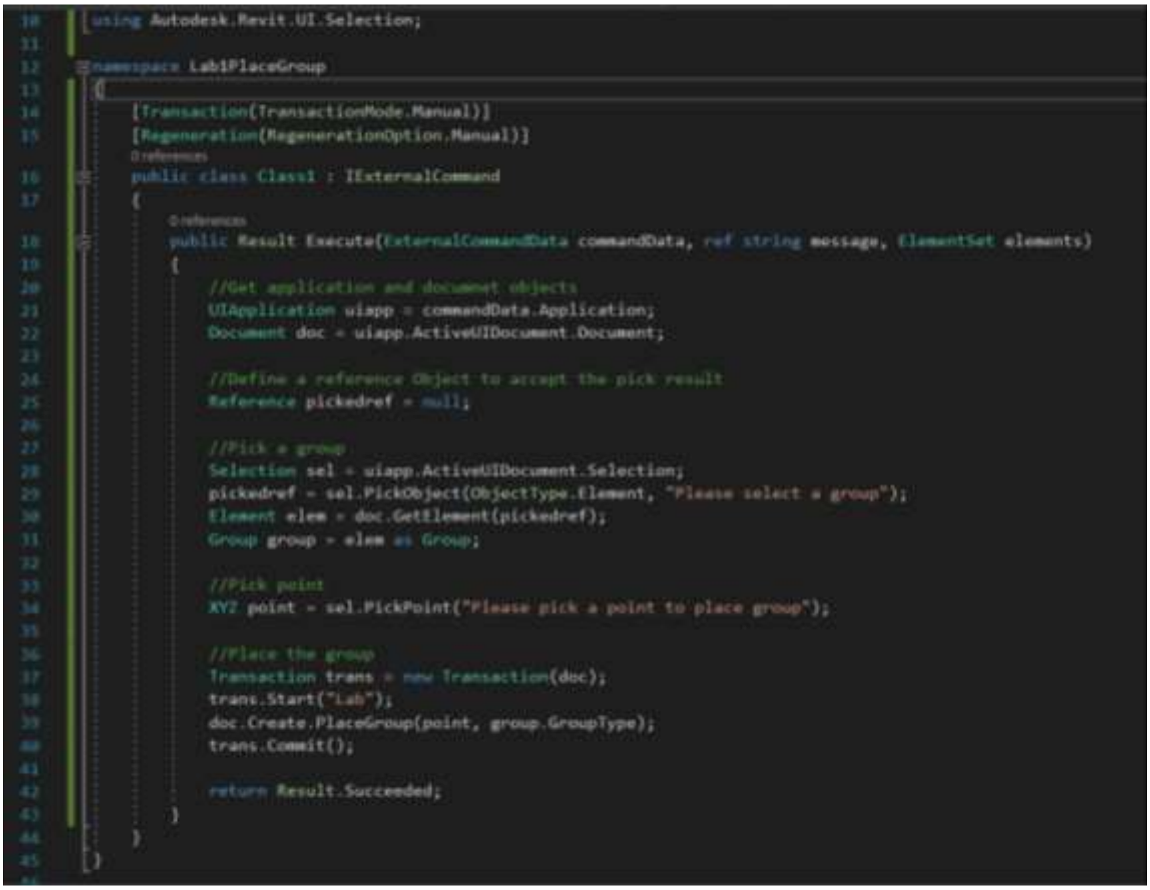
It is necessary now to add two reference files from the root Revit folder. Those are RevitAPI.dll, RevitAPIUI.dll. Add them both and now two interface DLL files are referenced in this current project. All the Revit APIs are exposed by these interface files and your project can use all of those available APIs from them [17].

6. Set the referenced files' Copy Local property value

In the Solution Explorer window you saw in step 5, click RevitAPI under Reference. In the Properties window, click Copy Local property, and then click the drop-down list and select False. Repeat the same steps to change the value of the RevitAPIUI.dll Copy Local property to False [17].

7. Add the code

Double click Class1.cs in the Solution Explorer window to show the code-editing window. Delete everything in this window and then type the following C# code. To get the full experience of developing with Visual Studio – including the use of features such as IntelliSense – we recommend you type the code from this guide rather than copying and pasting it. That said, if constrained for time you can also copy and paste into the Visual Studio code window: although this reduces the experience you gain from working with the code directly. An example of command class is shown in the figure 3.4:



```
using Autodesk.Revit.UI.Selection;

namespace Lab1PlaceGroup
{
    [Transaction(TransactionMode.Manual)]
    [Regeneration(RegenerationOption.Manual)]
    0 references
    public class Class1 : IExternalCommand
    {
        0 references
        public Result Execute(ExternalCommandData commandData, ref string message, ElementSet elements)
        {
            //Get application and document objects
            UIApplication uiapp = commandData.Application;
            Document doc = uiapp.ActiveUIDocument.Document;

            //Define a reference Object to accept the pick result
            Reference pickedref = null;

            //Pick a group
            Selection sel = uiapp.ActiveUIDocument.Selection;
            pickedref = sel.PickObject(ObjectType.Element, "Please select a group");
            Element elem = doc.GetElement(pickedref);
            Group group = elem as Group;

            //Pick point
            XYZ point = sel.PickPoint("Please pick a point to place group");

            //Place the group
            Transaction trans = new Transaction(doc);
            trans.Start("Lab");
            doc.Create.PlaceGroup(point, group.GroupType);
            trans.Commit();

            return Result.Succeeded;
        }
    }
}
```

Fig. 3.4. Code of creating start plug-in module

using System;

using System.Collections.Generic;

using System.Linq;

```
    using System.Text;
    using System.Threading.Tasks;
    using Autodesk.Revit.ApplicationServices;
    using Autodesk.Revit.Attributes;
    using Autodesk.Revit.DB;
    using Autodesk.Revit.UI;
    namespace DiplomaPlaceGroup{
    [Transaction(TransactionMode.Manual)]
    public class DiplomaRevitClass : IExternalCommand {
    public Result Execute(ExternalCommandData commandData, ref string message,
ElementSet elements {
    UIApplication uiApp = commandData.Application;
    UIDocument uiDoc = uiApp.ActiveUIDocument;
    uiApp.Application.FailuresProcessing += Application_FailuresProcessing;
    Autodesk.Revit.ApplicationServices.Application app = uiApp.Application;
    var uiDocumentInitial = CreateDocument(uiApp);
    _doc = uiDocumentInitial.Document;
    string[] allPaths = Directory.GetDirectories(SelectedDirectory);
    var families = GetFamiliesToParse(allPaths);
    if (families.Count < 1) return Result.Cancelled;
    _supplementedFamilyList = new List<FamilyData>();
    AddSupplementedFamilies(families);
    while (_warningsCount != 0) {
    string uiDocPath = string.Empty;
    using (var uiDocument = CreateDocument(uiApp)){
    using (_doc = uiDocument.Document){
            var failedFamilies = new List<FamilyData>();
            ShowProgressWindow(_failedFamilyPaths.Count);
            foreach (var path in _failedFamilyPaths){
                var family = GetFamilyDataFromFile(path);
                failedFamilies.Add(family);}
            _failedFamilyPaths.Clear();
            _warningsCount = 0;
            AddSupplementedFamilies(failedFamilies);}
        uiApp.OpenAndActivateDocument(uiDoc.Document.PathName);
        _doc?.Close(false);
        uiDocPath = uiDocument.Document.PathName;}
    File.Delete(uiDocPath);}
    uiApp.Application.FailuresProcessing -= Application_FailuresProcessing;
    //return initial project
    uiApp.OpenAndActivateDocument(uiDoc.Document.PathName);
    var uiDocPathInit = uiDocumentInitial.Document.PathName;
    uiDocumentInitial.Document?.Close(false);
    return Result.Succeeded;}
```

1. Save the file: On theFile menu, click Save All.
2. Change the .NET Framework

In the Solution Explorer window on the right hand side of Visual Studio window, right-click on DiplomaRevitClass and select Properties.

In the application option, set target framework to .NET framework 4.7.

3. Build the project: The code you have written is in human readable form. To make the code readable by a computer, you will need to translate it or "build" it.
4. Inside Visual Studio, in the Debug menu, click Build Solution to compile and build your plug-in. Build Success message shows in status bar of the Visual Studio window if the code is successfully built.

3.2. Writing an AddIn Manifest

An AddIn manifest is a file located in a specific location checked by Revit when the application starts. The manifest includes information used by Revit to load and run the plug-in [17].

1. Add the manifest code: start Notepad.exe from the Windows Start menu or any suitable text editing program. Write the following plug-in load settings to the opened editor.

```
<?xml version="1.0" encoding="utf-8"?>
<RevitAddIns>
  <AddIn Type="Application">
    <Name>Application RevitFamilyBrowser</Name>
    <Assembly>RevitFamilyBrowser.dll</Assembly>
     <FullClassName>RevitFamilyBrowser.App</FullClassName>
     <ClientId>d51df3b4-5cb9-1106-b4b3-0cb9e99ec117</ClientId>
    <VendorId>IvanKramarenko2020</VendorId>
  <VendorDescription>Ivan Kramarenko Diploma</VendorDescription>
</AddIn>
</RevitAddIns>
```

Depending on what version you are using you may need to change the path here to match your DiplomaSample.dll location on your computer:

C:\DiplomaSample\ DiplomaSample \bin\Release\DiplomaSample.dll

2. Save the file: in the program’s File menu, click Enter DiplomaSample.addin in the File name box. Change Save as type to the All Files option (the file name may be freely chosen, but the file extension must be “.addin”). Browse to the following subfolder, and then click the Save button.

For Windows 7/8.1/10 - C:\ProgramData\Autodesk\Revit\Addins\201*\   (The ProgramData folder is hidden by default).

3. Load your plug-in into Revit and allow the plug-in to communicate with Revit: Inside Revit on the Add-Ins ribbon tab, click the External Tools drop-down list, then click DiplomaPlaceGroup. This will start your plug-in.

4. Work with the plug-in: Move the cursor over Room1 in the Revit building model. When the cursor is hovering over the furniture group, its bounding box should be highlighted as per the below picture, with a tooltip showing Model Groups : Model Group : Group 1. Click to select this furniture group (fig. 3.5) [17].

5. Pick a point in another room, for example inRoom 2. The center of the new group is the point you selected.

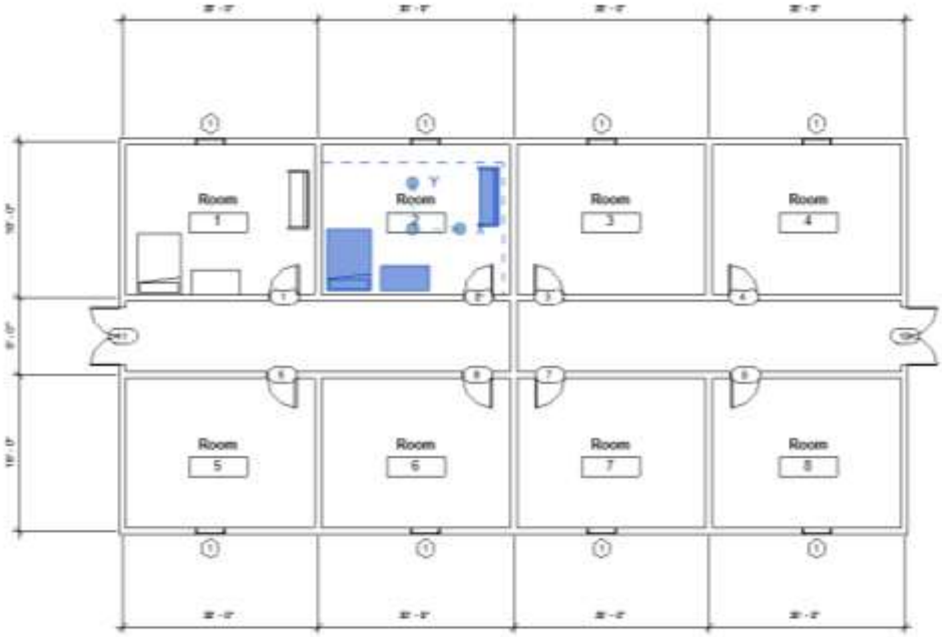


Fig. 3.5. The Revit building model

Now it is necessary to build the code [17]:

1. In the initial step, you just dispatched Visual Studio.

2. You at that point made another C# undertaking of type Class Library.

Since the improvement language utilized for this guide is C#, you are working with Visual Studio, and consequently you see Visual C# under Installed Templates part of the New Project discourse.

In the center part of this exchange, you saw different kinds of utilizations that can be made; you chose the layout as indicated by the sort of use you wish to make.

For modules to stack into Revit, they should be Class Library gatherings (DLLs). It's thus, in the subsequent advance, that you chose the Class Library layout. The name you entered is utilized to distinguish the undertaking inside the arrangement.

3. Your clear project was made, containing a couple of standard venture references to center .NET segments alongside a clear C# class document. It's this record that gets shown in the word processor window.

4. Saving the arrangement made actual records speaking to the substance of your group library venture on the PC's hard drive, permitting you to open and alter it at some other point later on.

5. This clear task, as made by Visual Studio, didn't naturally utilize the Revit API. For it to do as such, you added venture references to the interface DLLs in Revit portraying its API, dlland RevitAPIUI.dll.

6. When utilizing the Revit API, it is normal to add venture references to the two separate interface DLLs making up the API: one arrangements with center item usefulness, the other with the item's UI. You should connect your undertaking to these records to have the option to work with Revit API.

- dll contains the APIs to get to the Revit application, records, components, boundaries, and so forth
- dll contains the APIs identified with control and customization of the Revit UI, including order, determinations and exchanges

Having added your project references, it's significant that you set one of their properties suitably (fig. 3.6).

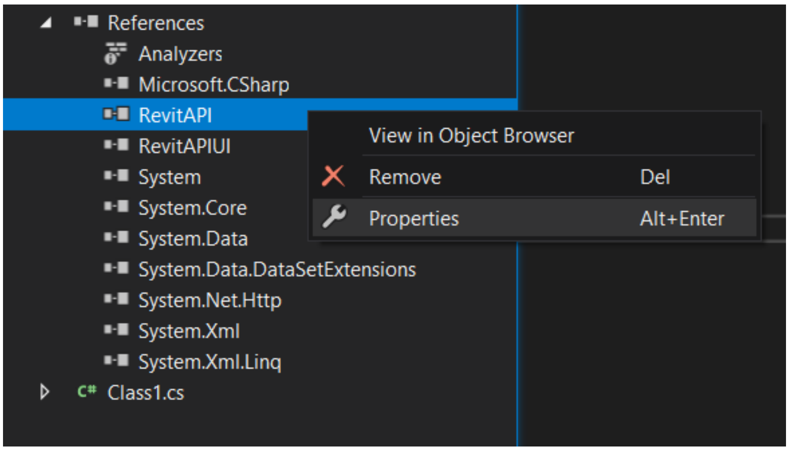


Fig. 3.6. The properties of RevitAPI

By default, Visual Studio adds project references with its Copy Local property set to True. This means that the referenced DLLs will get copied to the project’s output folder when it is built. In your case you wanted to change this setting to False, to make sure the DLLs did not get copied along with your assembly DLL [17].

7. Next you added C# code using the Revit API into your project. In other words providing Revit with instructions on how to perform the functionality of copying a user-selected group from one place to another.

Revit plug-ins are compiled into library assembly files (DLLs) which are then loaded and executed from within Revit’s memory space (fig. 3.7).

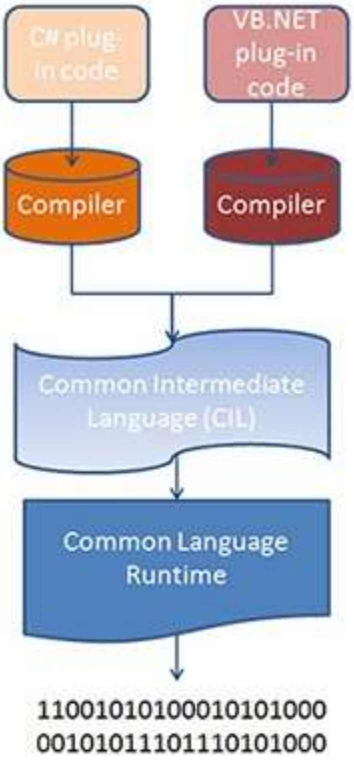


Fig. 3.7. Running Executables

When writing code, it is prudent to fabricate arrangements every once in a while to watch that there are no mistakes in the code. The code doesn't need to be finished or useful when constructing the arrangement. This methodology can help stay away from conceivably tedious investigating after code finish and has the side advantage of naturally saving altered yield documents prior to building.

During a .NET form, the CIL (what's in the fabricate) is gone through the CLR compiler as expected (JIT) to create its own (or machine's) code. JIT aggregation of CIL to local code happens at program startup. Since not all code is required at runtime, the JIT compiler turns CIL just when it is required, subsequently saving time and memory. It additionally stores any created code in memory, making it accessible for later use without the need to recompile.

In the last advance of this cycle, the PC's own code is executed by the processor.

3.3. Analyzing classes in a Revit plug-in

The Revit plugin class that implements this interface, known as the entry point for this plugin: it is the class, Revit try to find and call the Execute() method. In other words, when a Revit user clicks on a command in the Revit user interface listed under the External Tools drop-down menu button on the Add-ons tab, the code in the Execute () method is run (executed) with the corresponding class implementing this IExternalCommand interface.

```
[TransactionAttribute(TransactionMode.Manual)]public class Class1 : IExternalCommand{   public Result Execute(    ExternalCommandData commandData, ref string message,    ElementSet elements)  {    } }
```

Any block of code in a class that performs a particular task (or action) is called a method. In this case, the method declaration starts with the word public. You already know what public implies. For more information about understanding methods, see. In the advanced topics [11].

This method returns a result (actually Autodesk.Revit.UI.Result), not declared invalid (i.e. it returns nothing). The result you return from the Execute () method will show Revit whether the command succeeded or failed. If the command fails, any changes will be undone (Revit will roll back the transaction for which they were made).

The Execute() method has three parameters: commandData, message and elements. Let’s take a closer look at what each of these parameters refer to:

1. **commandData** is of type ExternalCommandData and provides a developer API access to the Autodesk Revit application. The application object in turn provides an access to the active document in Revit user’s interface and its corresponding database (i.e. all loaded elements, descriptions, families etc). All available Revit data (including model’s inforamtion) may be accessed by means of this commandData parameter.

2. **Message** is a string parameter with the additional ref keyword, which means it can be modified by the developer during the method execution. This parameter can be set in the external command even when the command fails or was cancelled. When this message is being set and the Execute method returns a failure or cancellation result, then an error dialog displayed by Revit application will show this message text.

3. **Elements** is a parameter of type ElementSet which allows developer to select elements to be highlighted on a screen if the command fails or was cancelled.

Now it is the time to take a look at the code inside in the Execute() method. This is the actual set of instructions which uses the Revit API to perform certain tasks when any command is being executed. This code just executes the same functionality, which can be made via Revit user interface [11].

Let’s look at the code, line-by-line:

// Get main application and all document objects from current UIApplication

UIApplication uiApp = commandData.Application;

In the first line is being used commandData parameter that was passed into the Execute() method to access the Application property of this object, which provides with access to the Revit application.

To have an opportunity to use the Application property just retrieved from the commandData parameter, it is needed to create a variable for the object named uiApp of type UIApplication. Then there was assigned the value of commandData.Application to this variable use in current command. Declaring a variable creates a named location for a value which can be accessed later on within the same block of code.

Document doc = uiApp.ActiveUIDocument.Document;

The uiApp variable (which contains the Revit Application object) provides access to the active document in the Revit user interface via the ActiveUIDocument property. In the above line of code – in just one line – you directly accessed the database of the active document (this database is represented by the Document class). In the current command this Document object was stored in a variable named doc.

Object Selection

The line below demonstrates how to prompt dialog window to select Groups using the RevitAPI.

Reference pickedRef = null;

It was begun by creating an empty variable named pickedRef of type Reference and was set its initial value to null. By means of this there was created an empty container in which is being stored a Reference object. Reference is a class which can contain elements from a Revit model associated with valid geometry [11].

Selection sel = uiApp.ActiveUIDocument.Selection;

pickedRef = sel.PickObject(ObjectType.Element, "Choose a group");

```
Element elem = doc.GetElement(pickedRef);
Group group = elem as Group;
```

Next there was accessed the current user selection. The user selection from the user interface is represented by the Selection property on the ActiveUIDocument object: you placed this Selection object into a variable named sel of type Selection. This Selection object provides a method named PickObject(). As the method's name suggests, it shifts focus to the user interface and prompts the user to select an object. The parameters of this method allow you to specify the type of element to be selected by the user (you can specify if you expect users to select faces, elements, edges, etc.) along with a message that the user will see in the bottom left corner of the Revit user interface while the plugin waits for the selection to take place [11].

Because the selected Group object has geometry data associated with it, it was safe to place it in the PickRef variable you declared earlier. You then used the Element reference property to access the associated reference element: in this case you assigned its value to a variable named elem of type Element. Because you expect the elem object to be of type Group, in the last line of the above code snippet you performed a "conversion", which allows us to treat the elem variable as a Group via a variable named group.

```
Element elem = doc.GetElement(pickedRef);
```

In the manufacturing scene, the term projecting alludes to the demonstration of setting a given material into a shape to shape it into an object of a specific structure. Additionally, in the programming scene, projecting methods the demonstration of attempting to set an estimation of one kind into another. Projecting requests that the language compiler think about an incentive in an alternate manner. In the last line of your code scrap, you are basically projecting the Element (which is really a Group chosen by the client) explicitly into the Group type. The as administrator in C# will make the compiler check the genuine sort of the article being projected: in the event that it is inconsistent with the objective kind, the worth returned by the administrator will be invalid [11].

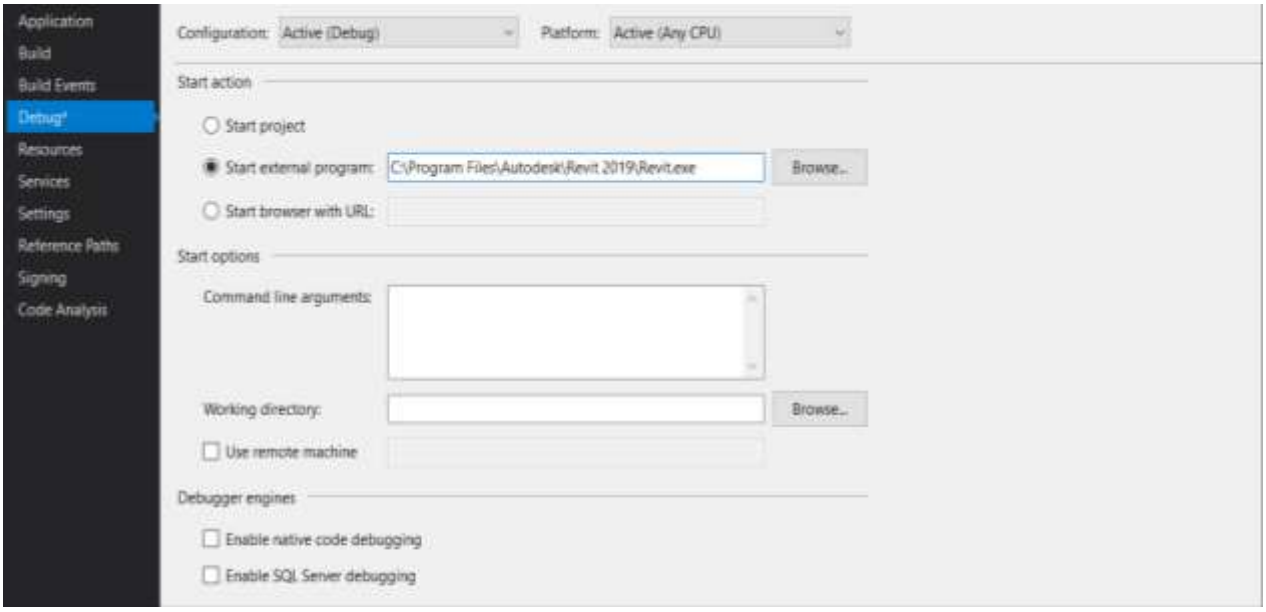
The point of this underlying module is to put a chose bunch at an area chose by the client. To play out this errand, there was utilized the PlaceGroup() technique from the dynamic record's information base item under the creation-related strategies made open through its Create property. This Create property makes it conceivable to add new examples of components -, for example, Groups - to the Revit model. The PlaceGroup() technique, true to form, expected you to pass in the area at which you needed to put your gathering, just as the sort (utilized with regards to Revit, as opposed to C#) of the gathering chose by the client [11].

At last, the exchange was submitted utilizing the Commit() strategy. This guaranteed the progressions typified by the exchange were effectively kept in touch with the Revit model.

return Result.Succeeded;

As it is possible to recall, the Execute() method is the entry point for a Revit plug-in – expects a Result to be returned. It is this Result which informs Revit whether the command completed successfully, whether it failed or was cancelled. At this point, assuming all went well with the code, there was passed back the Succeeded Result.

Thus the Properties settings allow you to set the executable to launch when you start the debugger (fig.3.8):



Configuration: Active (Debug)
Platform: Active (Any CPU)

Start action

○ Start project
● Start external program: C:\Program Files\Autodesk\Revit 2019\Revit.exe
○ Start browser with URL:

Start options

Command line arguments:
Working directory:
□ Use remote machine

Debugger engines

□ Enable native code debugging
□ Enable SQL Server debugging

Fig. 3.8. Properties of debug

To debug any plug-in code, it is necessary to launch Revit from the Visual Studio debugger and load needed plug-in into Revit.

In order to debug application using Visual Studio, user has to make some manual edits to the project files by implementing the following steps

1. In the Solution Explorer window, right mouse click on the project name (ProjectPlaceGroup) and select Properties to open new window with all project properties including Debug (fig. 3.8).
2. In the Properties window, Go to Debug option and choose “Start external program”. This menu allows deleloper to start certain program or file along with the application start to attach debugger directly.
3. Browse for “Revit.exe” with the help of windows file dialog. By the default the executable file of Revit application is located present in C:\Program Files\Autodesk\Revit 201x\Program\Revit.exe.

3.4. Launching the debugger

Before launching the debugger, guarantee that the AddIn show document for this lab has been made and exists at the area portrayed in the Writing an AddIn. Since client will currently be troubleshooting the module code, Visual Studio has made a 'investigate' rendition of your .NET module DLL.

To begin your investigating meeting, essentially open the Debug menu and select the Start Debugging choice, or just hit F5. This will dispatch Revit from your debugger. The kind of Revit that gets dispatched will rely upon the way you have given in the tag in the .csproj document that we altered physically [11].

It's important to dispatch Revit along these lines so a debugger can guide into the Revit cycle to screen the execution of the code that was ordered into a module DLL. Since there was set the AddIn show in the right area, Revit will consequently stack our module.

At the point when it is being builded a 'last' adaptation of a module DLL that you need to provide for your clients and clients, Visual Studio will assemble a delivery form. Visual Studio makes different enhancements to the assembled code in a delivery fabricate with the goal that it will run quicker and take up less memory. To investigate any code, Visual Studio will make a troubleshoot rendition of your module DLL. The troubleshoot form isn't advanced for speed/memory, and furthermore incorporates extra data that the debugger uses to educate client regarding what's going on when the code runs [11].

Presently, open the Project record.

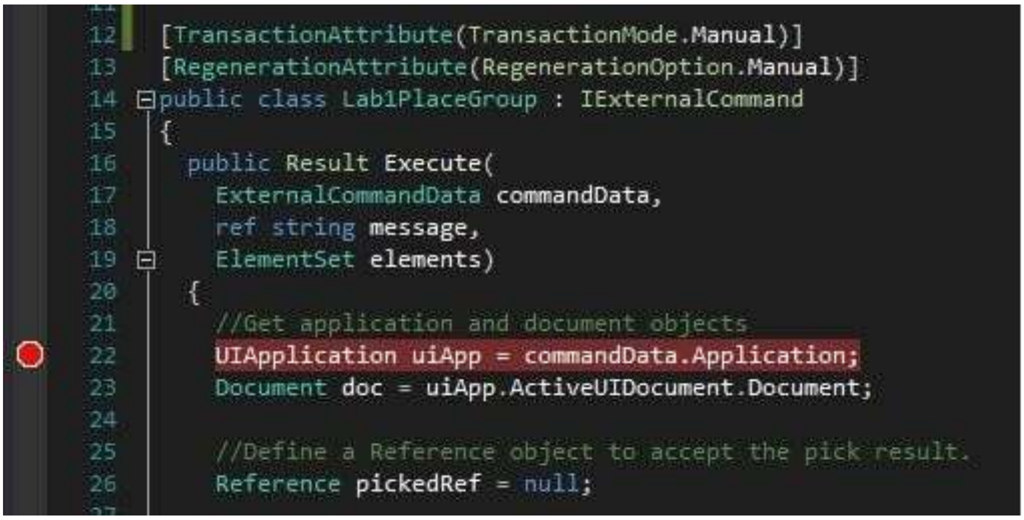
A module DLL is presently prepared to investigate. However, prior to running a DiplomaRevitClass order, it is important to advise the debugger to stop when it is executing your code. Do this utilizing a breakpoint.

In Visual Studio, double tap on Class1.cs in the Solution Explorer to show your code and snap anyplace in the line:

UIApplication uiApp = commandData.Application;

At that point select Toggle Breakpoint from the Debug menu (or hit F9).

That line will presently be featured in red and have a red hover in the edge close to it to show that there was set a breakpoint for this line (fig. 3.9).



```
[TransactionAttribute(TransactionMode.Manual)]
[RegenerationAttribute(RegenerationOption.Manual)]
public class Lab1PlaceGroup : IExternalCommand
{
  public Result Execute(
    ExternalCommandData commandData,
    ref string message,
    ElementSet elements)
  {
    //Get application and document objects
    UIApplication uiApp = commandData.Application;
    Document doc = uiApp.ActiveUIDocument.Document;

    //Define a Reference object to accept the pick result.
    Reference pickedRef = null;
```

Fig. 3.9. Breakpoint highliting in Visual Studio 2019

Set a breakpoint for the PickPoint function in the same way:

XYZ point = sel.PickPoint("Please pick a point to place group");

When Revit calls these methods in your code, the debugger will stop at these lines and wait for you to tell it what to do.

Stepping through your code [11]

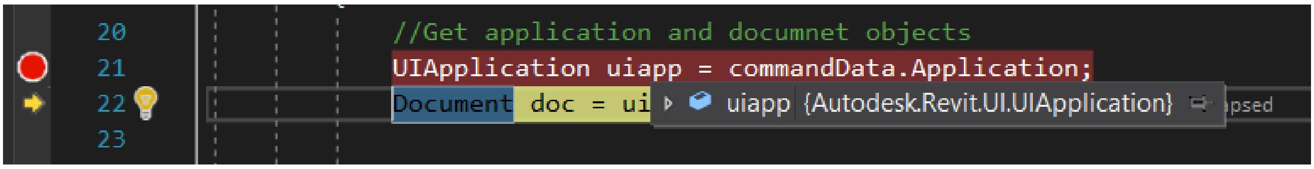
Now it’s time to invoke a command. Inside Revit on the Add-Ins ribbon tab, click the External Tools drop-down list, then click DiplomaPlaceGroup. This will start your plug-in in Revit and Visual Studio should take control and become a foreground application. (If it doesn’t, click on its icon in your Windows taskbar to activate it.) The debugger is now stopped waiting for you with the line of code it's about to execute highlighted in yellow and with a little yellow arrow in the margin next to it.

Now you're ready to step through your code. The Visual Studio debug menu gives you three ways to step through your code: Step Into; Step Over; and Step Out. You'll mostly be using Step Over – this executes the next line of code (the line highlighted in yellow) in the debugger, and then moves to the next line. If the line of code to be executed is a method call, then Step Over executes the entire method. If you also want to execute the called method a line at a time, you can use Step Into; and you can use Step Out to move back up (out of the method) to the code you were originally debugging [11].

As well as in the Debug menu, you should also see Step Into, Step Over and Step Out icons on a toolbar, and each also has a corresponding hot key (F11, F10, and Shift+F11).

Click on the Step Over icon on the toolbar now. The debugger moves to the next line of code (it ignores comments).

You can now hover the mouse over the text uiApp and a tooltip is displayed showing the value of the variable (fig. 3.10).



```
//Get application and documnet objects
UIApplication uiapp = commandData.Application;
Document doc = ui ▸ uiapp {Autodesk.Revit.UI.UIApplication}
```

Fig. 3.10. Right mouse click for the value of the variable

You can click on the sign to drill down to the various properties and check their values.

To display variable values, you can also right click on the variable in your code and select Add Watch. This displays the variable and its value in the Watch window at the bottom of your IDE.

Keep stepping through the code, hovering over variable values and properties to see how they change as the code executes. If at any point you want to stop line-by-line debugging, just hit F5 to Continue to the next breakpoint you had set – which in our case was set to the sel.PickPoint() line.

Stepping through this line by clicking F10, makes Revit the foreground application and prompts you to select the point where you want the Furniture group to be copied to. Let us select the approximate center of the adjacent room now. As soon as the point is selected, Visual Studio now becomes the foreground application with the next line of code highlighted in yellow with the debugger waiting for us to either check the value of the selected point or continue line by line debugging or just jump to the next breakpoint by clicking F5. At this point, we can check the XYZ value of the selected point by just hovering the mouse over the point variable (which is the container/variable we created to store the target point for the furniture group to be copied to) [11].

If you hit F5 now, you will find that the debugger steps through all the remaining code, and makes Revit the foreground application again with the copy operation successfully completed.

Once you've finished experimenting, select Stop Debugging from the Debug menu to close Revit and end your debugging session.

3.5. Family Browser Plug-in and Its Infrastructure

Modern projects for the construction of buildings, different structures and entire city infrastructures take years and thousands of man hours. Numerous engineers of different qualifications (electrical, piping, air conditioning etc.), architectures and specialists make digital models of constructions to calculate its loads, render views and a lot of other vital tasks. This work is filled with routine actions that engages at least half of this time. Engineers arrange electrical equipment, ventilation, piping, heating elements and a lot of the rest manually. But the problem is that to get to, for example, a socket or a lamp, you must press a lot of buttons on different panels, find the desired family of elements through windows explorer and, after loading the whole family into the project, pull out the desired type from there. Family Browser allows you to do it in a few seconds and just in two mouse clicks with clarity and filtering of the desired type. This is a dynamic interface to control, store and locate Revit families and types. Family Browser stays up to date with any changes made in windows explorer. Ideal for any small or large practice no matter what flavor of Autodesk’s Revit you use. All families can be controlled from a central location allowing a BIM Manager to instantly make changes, adding groups, tabs or families.

3.5.1 The Features of The Developed Plugin

1) during the installation of the .msi package (which is also ready to use), keys are created in the registry dynamically to control versions and directories, i.e., all libraries, .exe files, local databases and .addin files, which Revit needs directly;

2) when we start the audit, the relevance of all files required for the plugin is checked i.e., versions from the server are compared with the versions from the registry of a current user. If there is any update, the user has an opportunity to download all necessary files (families, templates, ifc export, etc.). Downloading these files is not just the case, but using the self-made FamiliesDownloader.exe file, which connects with the server and then downloads only necessary archives, extracts everything from them and distributes its contents into the required directories.

3) to simplify and perceptibly speed up the design in Revit software using the panel. The panel contains types from families that are grouped into categories - family categories are buttons from the header (fig. 3.11), for example, cables, electrical appliances, safety sensors etc. These types can be double-clicked or dragged into the project without any extra effort of uploading family into the project and extract from there a type by yourself.

4) The insertion occurs by a query in the database, which stores all the info about families and its types (path to the family from which to insert it, path to the picture, description, name, installation type, installation place, etc.). The code of these particular operations is displayed in Appendix B.

The main panel of Revit Family Browser plug-in with all needed family types and various filters is displayed at the figure 3.12.

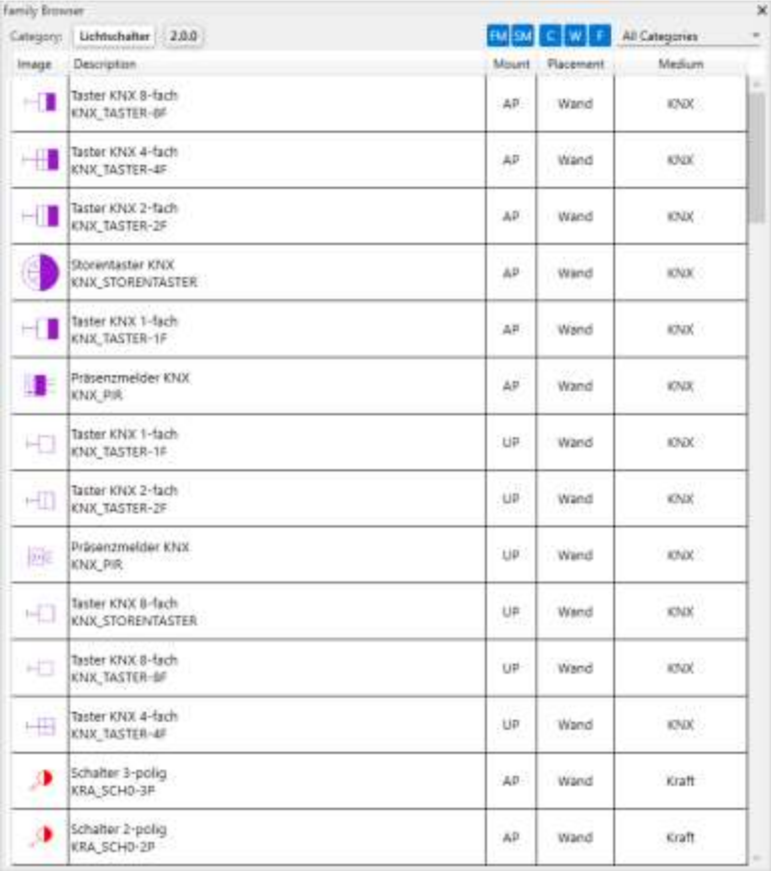


Image	Description	Mount	Placement	Medium
	Taster KNX 8-fach KNX_TASTER-8F	AP	Wand	KNX
	Taster KNX 4-fach KNX_TASTER-4F	AP	Wand	KNX
	Taster KNX 2-fach KNX_TASTER-2F	AP	Wand	KNX
	Storentaster KNX KNX_STORENTASTER	AP	Wand	KNX
	Taster KNX 1-fach KNX_TASTER-1F	AP	Wand	KNX
	Präsenzmelder KNX KNX_PIR	AP	Wand	KNX
	Taster KNX 1-fach KNX_TASTER-1F	UP	Wand	KNX
	Taster KNX 2-fach KNX_TASTER-2F	UP	Wand	KNX
	Präsenzmelder KNX KNX_PIR	UP	Wand	KNX
	Taster KNX 8-fach KNX_STORENTASTER	UP	Wand	KNX
	Taster KNX 8-fach KNX_TASTER-8F	UP	Wand	KNX
	Taster KNX 4-fach KNX_TASTER-4F	UP	Wand	KNX
	Schalter 3-polig KRA_SCHO-3P	AP	Wand	Kraft
	Schalter 2-polig KRA_SCHO-2P	AP	Wand	Kraft

Fig. 3.11. The main electrics panel of Family Browser

5) the panel on the right displays types from the families of a certain category, which are loaded by clicking on the category (fig. 3.15) from the database.

6) to ease navigating through current types there were created various types of filtering current family types, such as filtering by type / location / installation medium (fig. 3.12, fig. 3.13).

3.5.2 Project Standards Regulation with Family Browser

Another problem is that one project contains particular standards for families, types, annotations, templates and other BIM parts. But those standards are often being missed because a lot of engineers can’t wait till some special family will be designed for them right here and right now. Therefore Family Browser application has its own standard-checking pipeline to create, check and push any new family type to standardized database on cloud via AWS instruments. The overview of this method is displayed below:

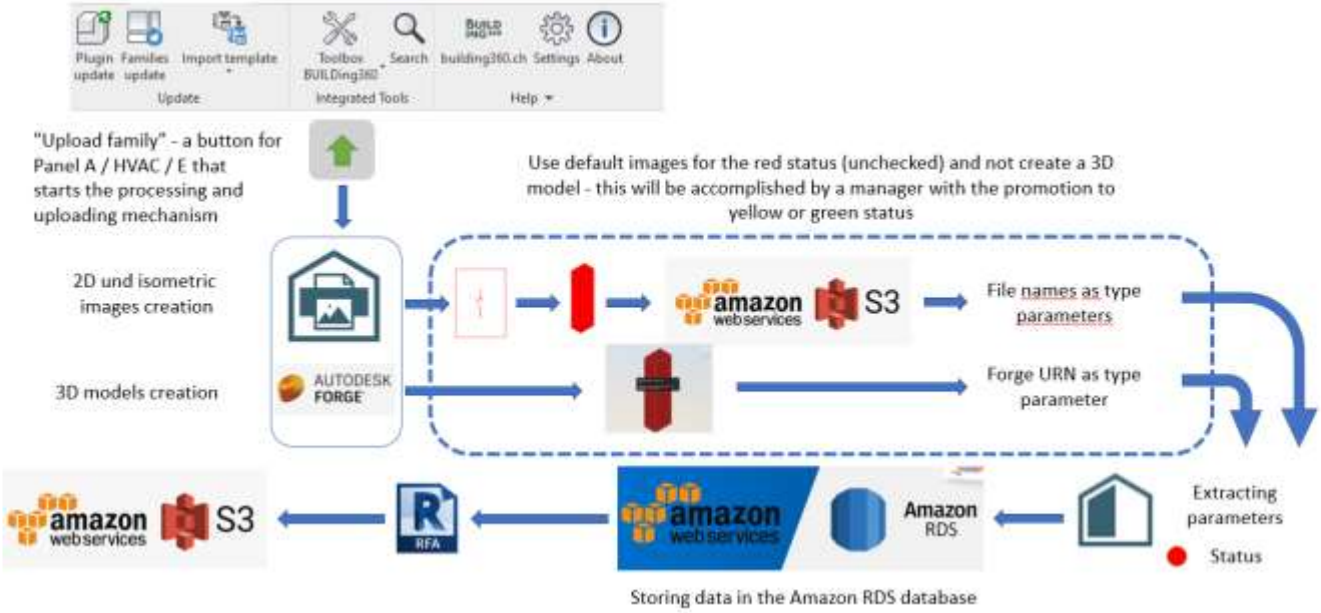

Plugin update
Families update
Import template
Update
Toolbox BUILDing360
Search
Integrated Tools
building360.ch
Settings
About
Help
"Upload family" - a button for Panel A / HVAC / E that starts the processing and uploading mechanism
Use default images for the red status (unchecked) and not create a 3D model - this will be accomplished by a manager with the promotion to yellow or green status
2D und isometric images creation
3D models creation
AUTODESK FORGE
amazon webservices
S3
File names as type parameters
Forge URN as type parameter
RFA
Amazon RDS
Extracting parameters
Status
Storing data in the Amazon RDS database

Fig. 3.12. Family submission process for standards in FamilyBrowser

This pipeline can be described as a bunch of simple steps for engineer, but overall it's a huge infrastructure which consists of Revit plug-ins and Amazon Web Services. These mechanism is performed in the following way:

1) There is a tempate for each category in Revit on the base of which engineer creates its own family for project. Template is standardized and it is stored in AWS S3 to make it always up to date and easy to download (fig. 3.13).

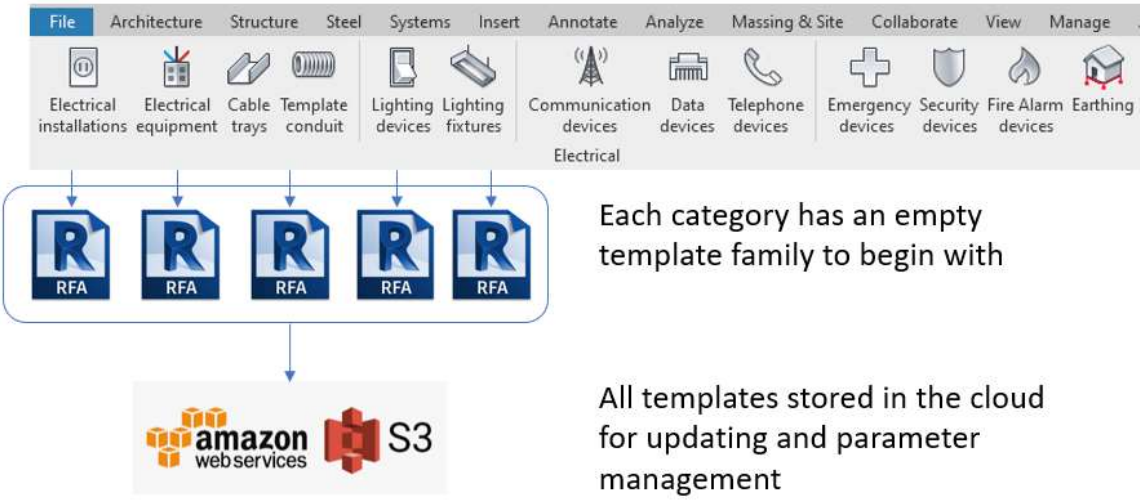


Fig. 3.13. First step of creating own standardized family

2) After family is created it must be processed by internal FamilyBrowser script, which retrieves all internal parameters of each type and to store in in Amazon RDS. Then starts another Revit addin, called Image Printer. This addin creates images of each type from family automatically. and uploaded to AWS S3. After all new family is being uploaded to AWS S3 as .rfa file.

3) Next step is all responsible head-engineers and managers receive email or another type of notification that the family is created and its unconfirmed yet (red status). Also, this family is already available in Revit Family Browser application, but with red status.

4) Managers and head-engineers have access to read, modify, delete all families including those unconfirmed. They decide what to do with new family and then set status to yellow (checked) or green (checked and released) or delete family (fig 3.14).

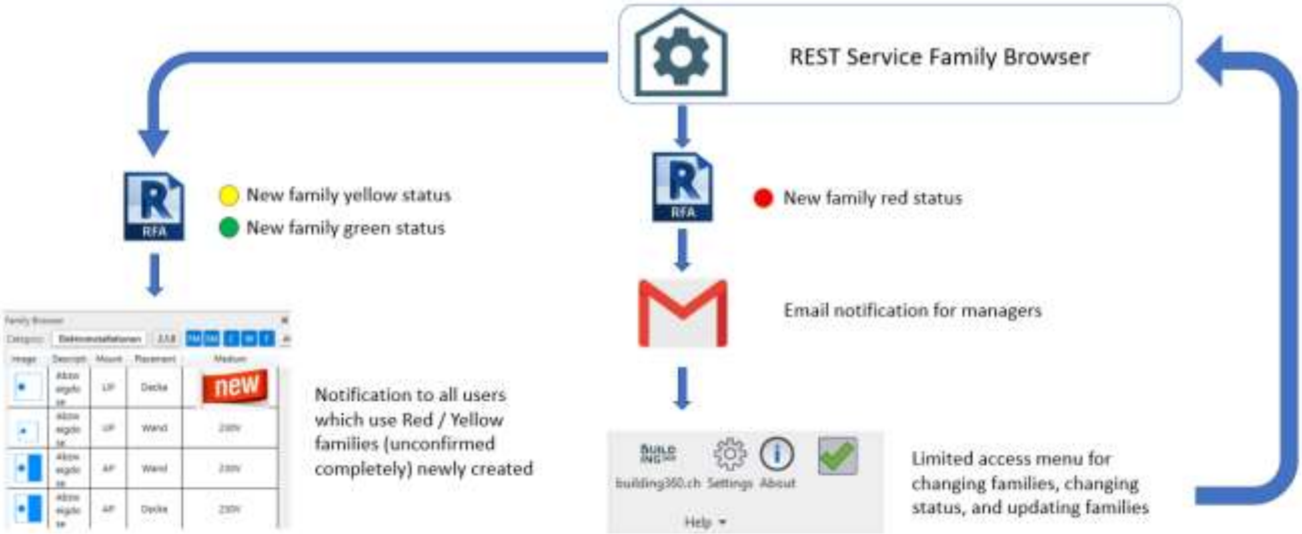

REST Service Family Browser
RFA
New family yellow status
New family green status
RFA
New family red status
Email notification for managers
new
Notification to all users which use Red / Yellow families (unconfirmed completely) newly created
building360.ch Settings About
Help
Limited access menu for changing families, changing status, and updating families

Fig. 3.14. Family submition process and representation in Revit

3.5.3 Family Browser Interface and Basic Functionality Overview

All families are grouped by category. Once category is selected all available families will be displayed in the main panel (fig. 3.15-3.16) [18].

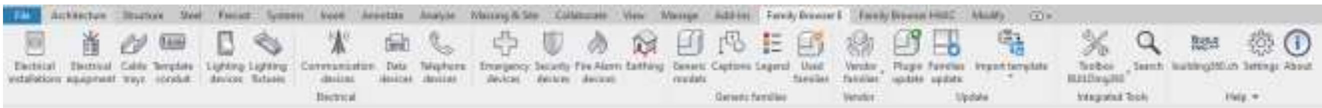


Fig. 3.15. Electrics tab integrated in Revit default tabs with its ribbon panel

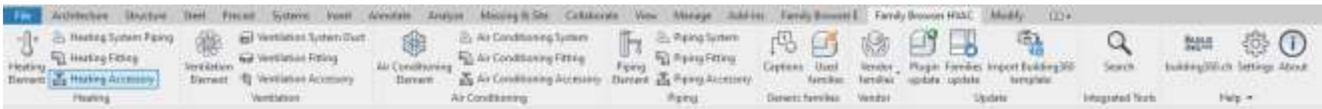


Fig. 3.16. HVAC tab with its own ribbon panel

Main Panel can be incorporated to the existing Revit panels such as "Parameters" or "Project Browser". This panel contains filters, list of available types and allows to load them into project (fig.3.11).

Current category name is displayed in the "Category" field on top of the table along with the current selected version.

There are three main filters and one optional for element based on their internal parameters:

- Flush Mount / Surface Mount;
- Placement on Ceiling / Wall / Floor;
- Installation Medium (list of systems depends on selected system);

Search bar (fig. 3.17) which filters current family types by your input string (available in Used Families, Families Update, Vendor Families).

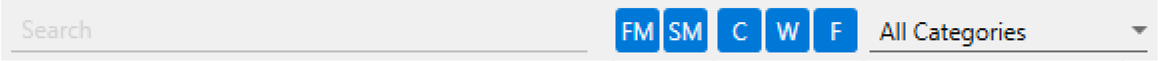


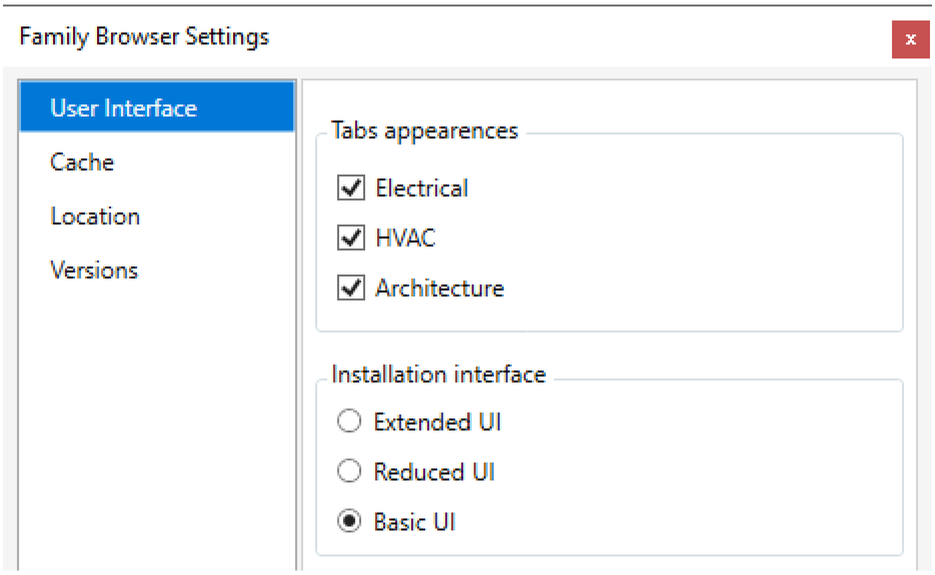
Fig. 3.17. Family Browser dockable panel filtering header

Once selection filter is activated, related items and items without such parameter will be displayed in a list. Multiple filters may work together - you can choose Flush Mount elements for placing on Wall for 400V system. Filters are toggle buttons, i.e. to unselect filter - click it again.

Settings

Settings allow changing the user interface, language and families version [18].

- User Interface Settings & Installation interface (fig. 3.18) - on the "User Interface" tab you can set visible three Electrical, HVAC, Architecture panels or only one of them. Changes implemented on the fly without restarting Revit application.



Tabs appearences

☑ Electrical

☑ HVAC

☑ Architecture

Installation interface

○ Extended UI

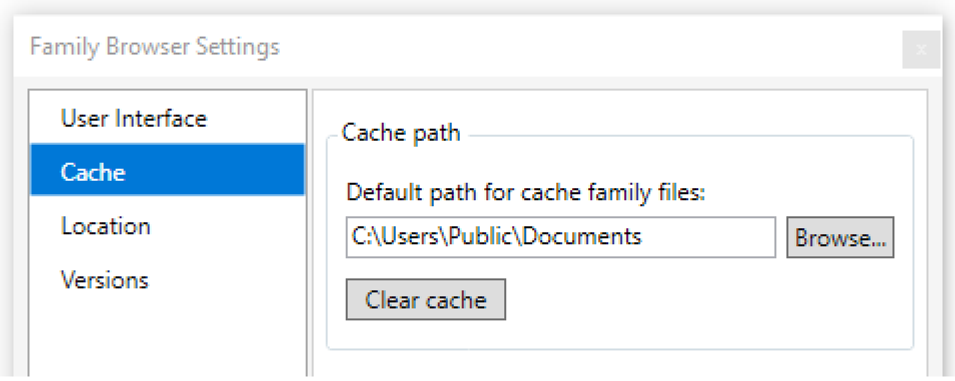
○ Reduced UI

● Basic UI

Fig. 3.18. User Interface Settings tab

On this tab it is also available to change installation interface, which appears during Family Browser add-in updates. It is possible to select whether control installation directory and other default Windows .msiЫ package installation options.

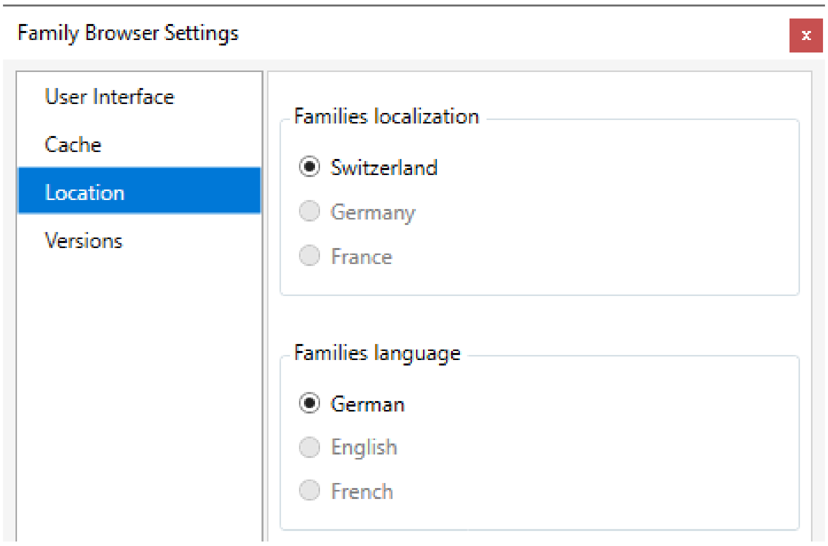
- Cache Settings - on the "Cache" tab (fig. 3.19) you can set folder with family files, its images, configurations, and other data for Revit Family Browser and clear it to free space on your disk drive.



Default path for cache family files: C:\Users\Public\Documents

Fig. 3.19. Cache Settings tab

- Location Settings - on the "Location" tab (fig 3.20) you can switch between available content (i.e. family files, types, templates) localizations and languages.



Families localization

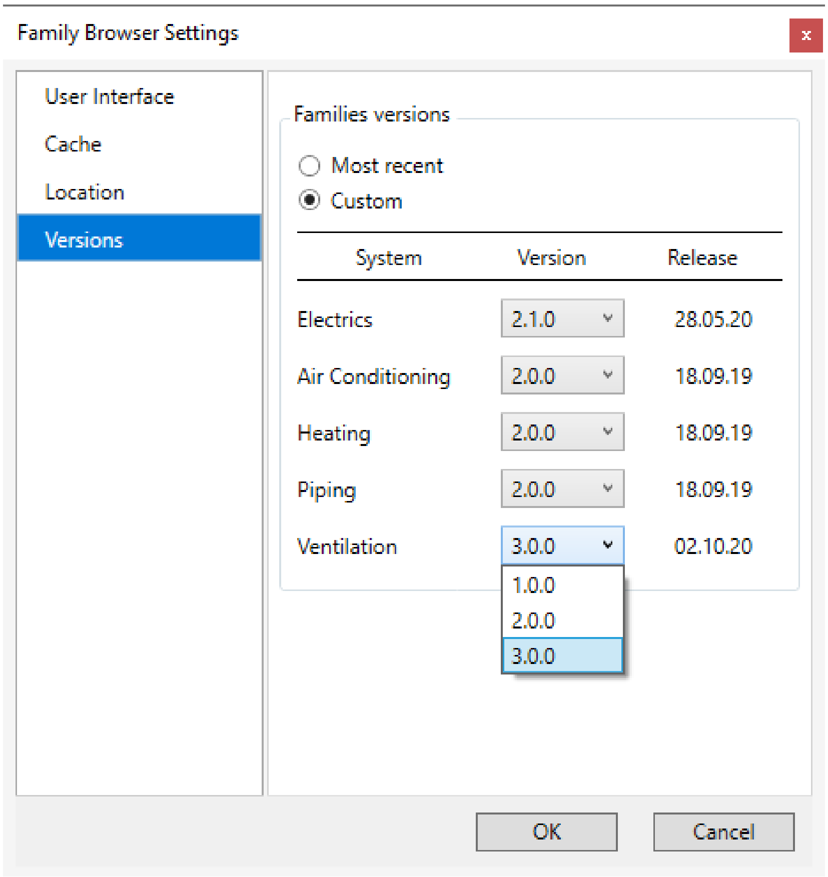
● Switzerland
○ Germany
○ France

Families language

● German
○ English
○ French

Fig. 3.20 Locations Settings tab

- Version Settings - on the "Versions" tab (fig 3.21) it is possible to change families' version separately for each system you want to use in your project. For all versions you can see families' release date and brief description of this version as a tooltip on dropdown menu. Also there is a tool that gives an opportunity to change any of standardized families that were used in current project to version selected in this settings tab.



Families versions

○ Most recent
● Custom

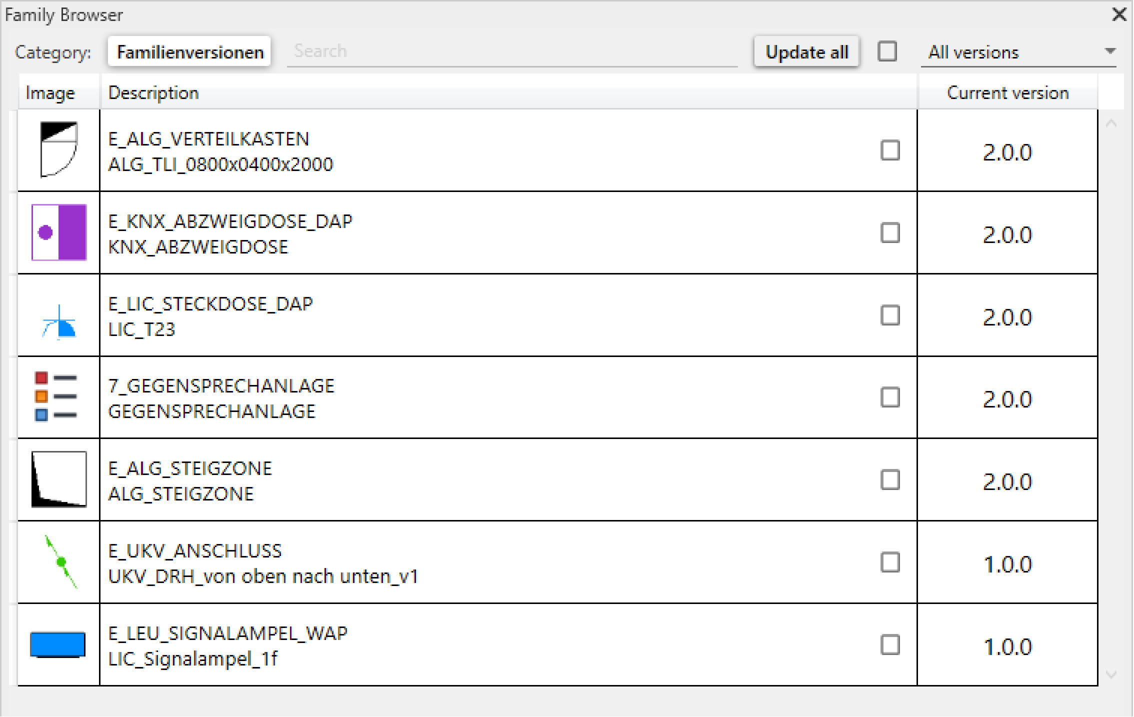
System	Version	Release
Electrics	2.1.0	28.05.20
Air Conditioning	2.0.0	18.09.19
Heating	2.0.0	18.09.19
Piping	2.0.0	18.09.19
Ventilation	3.0.0	02.10.20

Fig. 3.21. Versions Settings tab

It is recommended to use the latest families' version, unless your project is made with older version and you do not wish to have extra parameters. User's custom properties will be transferred during the process of families version change [18].

3.5.4 Updating Family Browser Content

To change families' version within project you need to press the button with an arrow. This mechanism will collect all used family types in the current project and display them in the main Family Browser dockable panel. The collection process may take some time in large projects, but it can be cancelled anytime. Each family type, created by BUILDing360 has a preview image, description and version (fig. 3.22a). You can filter Families by version using drop-down filter (fig. 3.22b) or via internal parameters in the search bar (fig. 3.22c).

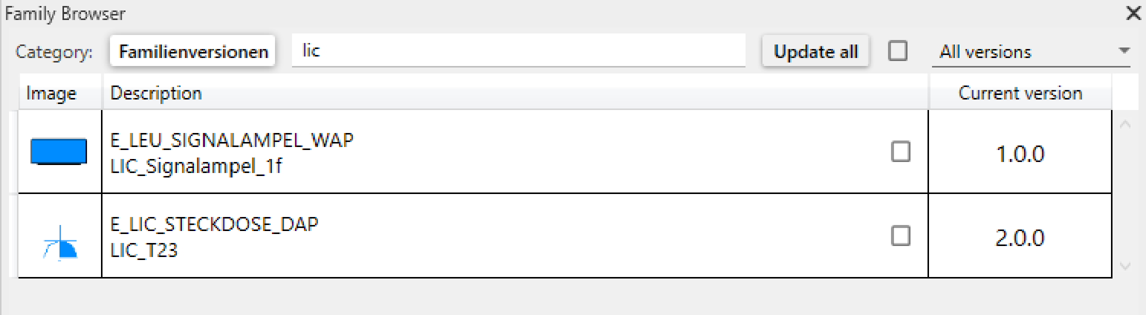


Family Browser

Category: Familienversionen Search Update all ☐ All versions

Image	Description		Current version
	E_ALG_VERTEILKASTEN ALG_TLI_0800x0400x2000	☐	2.0.0
	E_KNX_ABZWEIGDOSE_DAP KNX_ABZWEIGDOSE	☐	2.0.0
	E_LIC_STECKDOSE_DAP LIC_T23	☐	2.0.0
	7_GEGENSPRECHANLAGE GEGENSPRECHANLAGE	☐	2.0.0
	E_ALG_STEIGZONE ALG_STEIGZONE	☐	2.0.0
	E_UKV_ANSCHLUSS UKV_DRH_von oben nach unten_v1	☐	1.0.0
	E_LEU_SIGNALAMPEL_WAP LIC_Signalampel_1f	☐	1.0.0

Fig. 3.22a. Collected family types from the project via Family Browser tool

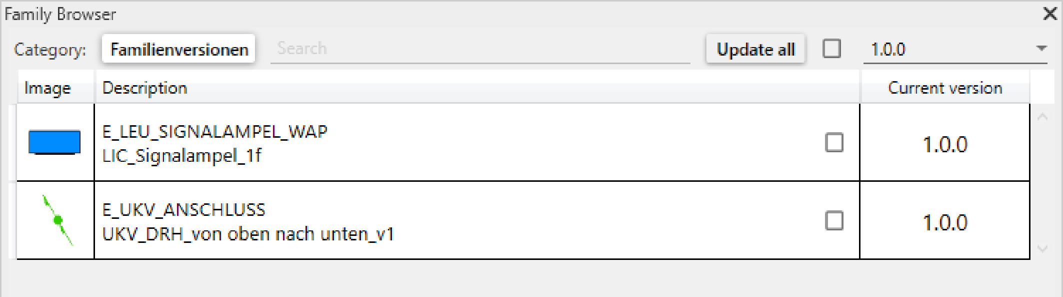


Family Browser

Category: Familienversionen lic Update all ☐ All versions

Image	Description		Current version
	E_LEU_SIGNALAMPEL_WAP LIC_Signalampel_1f	☐	1.0.0
	E_LIC_STECKDOSE_DAP LIC_T23	☐	2.0.0

Fig. 3.22b. Filtered family types by description



Family Browser

Category: Familienversionen Search Update all ☐ 1.0.0

Image	Description		Current version
	E_LEU_SIGNALAMPEL_WAP LIC_Signalampel_1f	☐	1.0.0
	E_UKV_ANSCHLUSS UKV_DRH_von oben nach unten_v1	☐	1.0.0

Fig. 3.22c. Filtered family types by version number

First of all select families version you want to use in Settings.

By pressing "Update all" or by selecting checkbox near this button and then pressing it - you can change version of all families within current project.

All custom properties will be transferred automatically during update process and they will not be lost.

Families version may be either upgraded or downgraded. Simply set preferred version in Settings and update it in “Families update”.

Present types in families may vary between different versions. If you used version 1.0.0 and can't find a specific type in version 2.0.0 - you have to set version 1.0.0 in Settings or use another elements [18].

3.6. FamilyBrowser REST Service via Laravel

3.6.1 What is Laravel

Laravel (fig.3.23) is a PHP framework developed with PHP developer productivity in mind. Written and maintained by Taylor Otwell, the framework is very opinionated and strives to save developer time by favoring convention over configuration. The framework also aims to evolve with the web and has already incorporated several new features and ideas in the web development world—such as job queues, API authentication out of the box, real-time communication, and much more [20].

Fig. 3.23. Laravel framework logo

3.6.2 RESTful APIs and its description

First, it is necessary to understand what exactly is considered a RESTful API. REST stands for REpresentational State Transfer and is an architectural style for network communication between applications, which relies on a stateless protocol (usually HTTP) for interaction.

In RESTful APIs, we use the HTTP verbs (fig.3.24) as actions, and the endpoints are the resources acted upon. There are such HTTP verbs for their semantic meaning:

- GET: retrieve resources
- POST: create resources
- PUT: update resources

- DELETE: delete resources

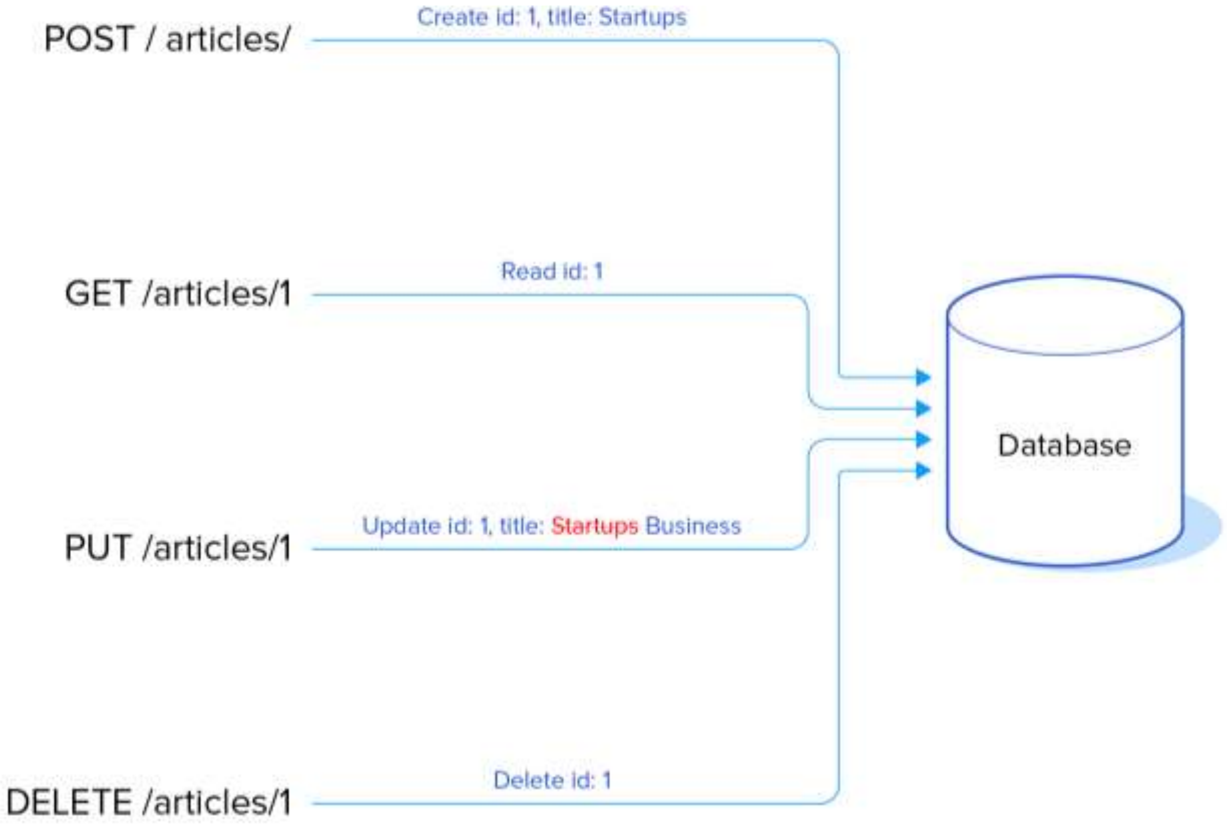


Fig. 3.24. HTTP requests to database

Update Action: PUT vs. POST

RESTful APIs are a matter of much debate and there are plenty of opinions out there on whether is best to update with POST, PATCH, or PUT, or if the create action is best left to the PUT verb. In this article we’ll be using PUT for the update action, as according to the HTTP RFC, PUT means to create/update a resource at a specific location. Another requirement for the PUT verb is idempotence, which in this case basically means you can send that request 1, 2 or 1000 times and the result will be the same: one updated resource in the database [20].

Resources

Resources will be the targets of the actions, in our case Articles and Users, and they have their own endpoints:

- /articles
- /users

In this laravel API description, the resources will have a 1:1 representation on our data models, but that is not a requirement. It is allowed to have resources represented in more than one data model (or not represented at all in the database) and models completely off limits for the develpoer. In the end, it is not prohibited to decide how to architect resources and models in a way that is fitting to any kind of application.

Setting Up a Laravel Web Service Project

As with all modern PHP frameworks, it is needed Composer to install and handle our dependencies. After you follow the download instructions (and add to your path environment variable), install Laravel using the command [20]:

```
$ composer global require laravel/installer
```

After the installation finishes, you can scaffold a new application like this:

```
$ laravel new myapp
```

For the above command, you need to have ~/composer/vendor/bin in your $PATH. If you don’t want to deal with that, you can also create a new project using Composer:

```
$ composer create-project --prefer-dist laravel/laravel myapp
```

Migrations and Models

Before writing the first migration, make sure you have a database created for this app and add its credentials to the .env file located in the root of the project.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=homestead
DB_USERNAME=homestead
DB_PASSWORD=secret
```

Let’s get started with our first model and migration—the Article. The article should have a title and a body field, as well as a creation date. Laravel provides several commands through Artisan—Laravel’s command line tool—that help us by generating files and putting them in the correct folders. To create the Article model, we can run:

```
$ php artisan make:model Article -m
```

The -m option is short for --migration and it tells Artisan to create one for our model. Here is the generated migration:

```
<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class FamilyTypeDataE extends Migration{
   /**
    * Run the migrations.
    * @return void
    */
   public function up() {
      Schema::create(' FamilyTypeDataE ', function (Blueprint $table) {
         $table->increments('id');
         $table->timestamps();
      }); }
   /**
    * Reverse the migrations.
    * @return void
    */
   public function down() {
      Schema::dropIfExists(' FamilyTypeDataE '); }}
```

Laravel out of the box comes with two migrations, create_users_table and create_password_resets_table. We won't be using the password_resets table, but having the users table ready for us will be helpful.

Now let's go back to our model and add those attributes to the $fillable field so that we can use them in FamilyTypeData::create and FamilyTypeData::update models:

```
class FamilyTypeData extends Model
{protected $fillable = ['title', 'body'];}
```

Routes and Controllers

Time to create the basic endpoints for our application: create, retrieve the list, retrieve a single one, update, and delete. On the routes/api.php file, we can simply do this:

```
<?php
namespace App\Http\Controllers;
```

```
use App\Models\FamilyTypeDataE;
use Illuminate\Http\Request;
class FamilyTypeDataEController extends Controller {
    public function index() {
        return FamilyTypeDataE::all();   }
    public function store(Request $request){
         FamilyTypeDataE::create($request->all());   }
    public function show(FamilyTypeDataE $familyTypeDataE) {
        return $familyTypeDataE;   }
    public function update (Request $request, $id){
        $type = FamilyTypeDataE::findOrFail($id);
        $type ->update($request->all());
        return $ type;  }
    public function delete(Request $request, $id)  {
        $ type = FamilyTypeDataE::findOrFail($id);
        $ type ->delete();
        return 204; }
}
```

That was just an example of numerous api paths from the real application, the list of some of them is below in the routes/api.php file:

- Route::get('Category/system/{systemName}', 'CategoryController@showBySystem');
- Route::get('FamilyData/category/{category}/system/{system}/version/{version}', 'FamilyDataController@showParams');
- Route::get('FamilyData/system/{system}/version/{version}', 'FamilyDataController@showVersion');
- Route::get('FamilyData/version/{version}/familyName/{familyName}', 'FamilyDataController@showByName');
- Route::get('FamilyData/params/{id}','FamilyDataController@getByParams');
- Route::get('FamilySystem', 'FamilySystemController@index');
- Route::get('FamilySystem/versions','FamilySystemController@getVersions');

- Route::get('FamilyTypeData/system/{system}/version/{version}/typeId/{typeId}', 'TypeDataController@getTypeById');
- Route::get('FamilyTypeData/system/{system}/version/{version}/familyName/{familyName}', 'TypeDataController@getTypeByFamily');
- Route::get('FamilyTypeData/system/{system}/version/{version}/familyName/{familyName}/typeName/{typename}', 'TypeDataController@getTypeByName');
- Route::get('FamilyTypeData/system/{system}/version/{version}/category/{category}', 'TypeDataController@getTypeByCategory');
- Route::get('FamilyTypeData/system/{system}/listDbId/{listDbId}', 'TypeDataController@getTypesByListDbId');
- Route::get('FamilyTypeData/system/{system}/version/{version}/search/{search}', 'TypeDataController@searchType');
- Route::get('FamilyTypeData/system/{system}/version/{version}/latestId', 'TypeDataController@getLatestId');
- Route::get('FamilyVersion', 'FamilyVersionController@index');
- Route::get('FamilyVersion/{id}', 'FamilyVersionController@show');
- Route::get('InstallationMedium', 'InstallationMediumController@index');
- Route::post('FamilyTypeData','TypeDataController@store');

A Note on HTTP Status Codes and the Response Format

There was added the response()->json() call to our endpoints. This lets us explicitly return JSON data as well as send an HTTP code that can be parsed by the client. The most common code that will be returning [20]:

- 200: OK. The standard success code and default option.
- 201: Object created. Useful for the store actions.
- 204: No content. When an action was executed successfully, but there is no content to return.
- 206: Partial content. Useful when you have to return a paginated list of resources.
- 400: Bad request. The standard option for requests that fail to pass validation.
- 401: Unauthorized. The user needs to be authenticated.

- 403: Forbidden. The user is authenticated but does not have the permissions to perform an action.
- 404: Not found. This will be returned automatically by Laravel when the resource is not found.
- 500: Internal server error. Ideally it is not going to be explicitly returning this, but if something unexpected breaks, this is what your user is going to receive.
- 503: Service unavailable. Pretty self-explanatory, but also another code that is not going to be returned explicitly by the application.

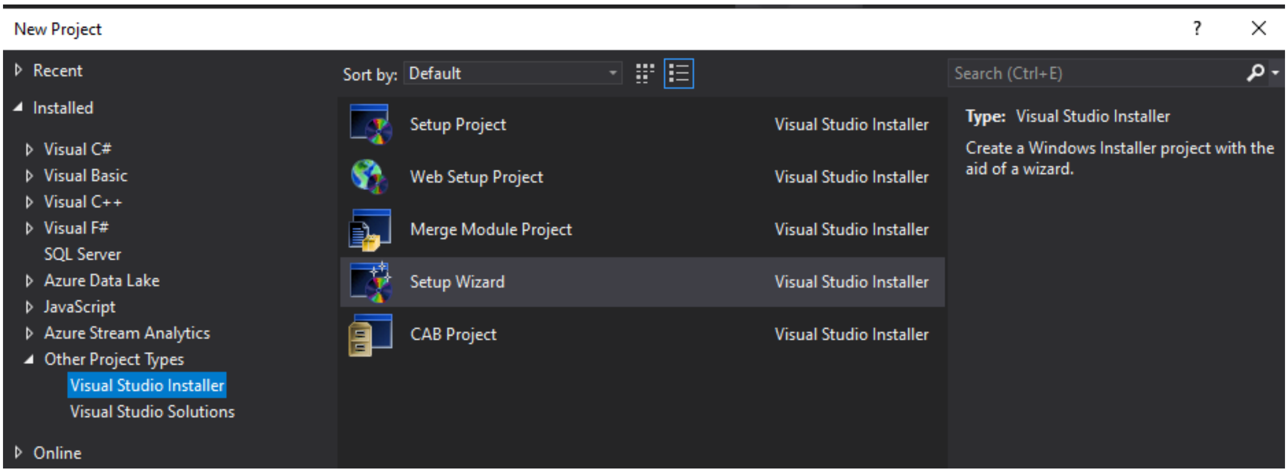
3.8 Creating Installer Package for Revit Family Browser Application

Visual Studio 2017 contains a bundle that empowers everybody to make Windows installer documents for your applications. These straightforward strides to construct Revit Family Browser own arrangement bundle for the Visual Studio.

3.8.1 Creating an installer within your existing Visual Studio project

Start by adding an installer project by right-clicking on your solution folder and choosing Add -> New Project.

Then in the Add New Project dialog choose Visual Studio Installer and Setup Wizard. Give your setup program a name and press OK (fig.3.25):



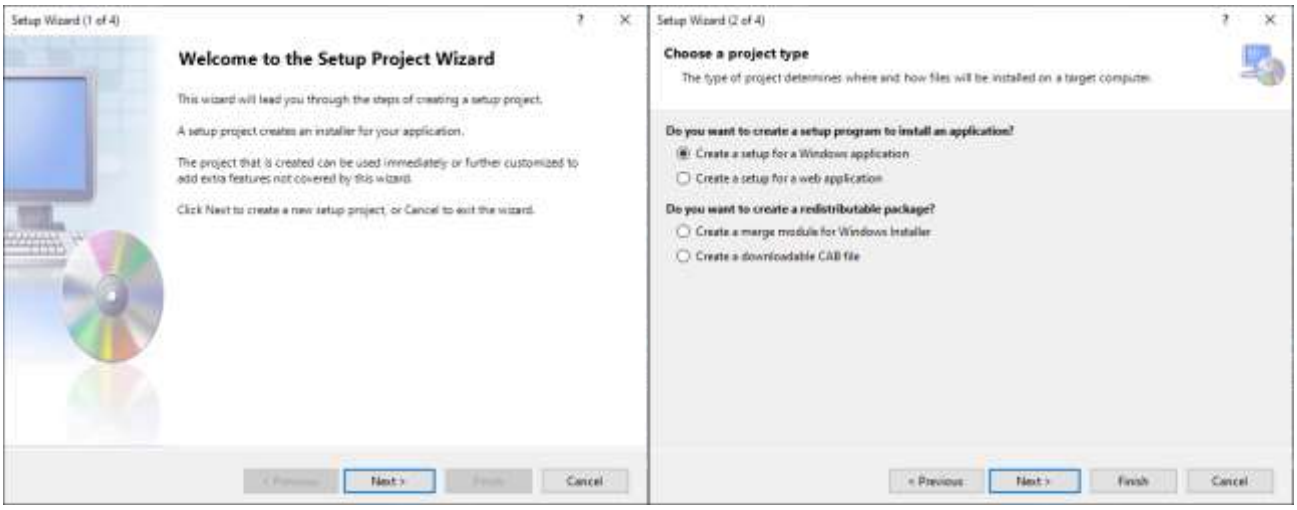
Setup Project	Visual Studio Installer
Web Setup Project	Visual Studio Installer
Merge Module Project	Visual Studio Installer
Setup Wizard	Visual Studio Installer
CAB Project	Visual Studio Installer

Type: Visual Studio Installer

Create a Windows Installer project with the aid of a wizard.

Fig. 3.25. Selecting project in Visual Studio 2017

The Setup Wizard (fig. 3.26) will then walk you through the remaining steps. First click next and then decide if you need a Windows or Web application. Commonly it is not yet needed any of the other special packages, so just click next. Even if you do, it is still possible add them later:


Setup Wizard (1 of 4)

Welcome to the Setup Project Wizard

This wizard will lead you through the steps of creating a setup project.

A setup project creates an installer for your application.

The project that is created can be used immediately or further customized to add extra features not covered by this wizard.

Click Next to create a new setup project, or Cancel to exit the wizard.

Setup Wizard (2 of 4)

Choose a project type

The type of project determines where and how files will be installed on a target computer.

Do you want to create a setup program to install an application?

● Create a setup for a Windows application

○ Create a setup for a web application

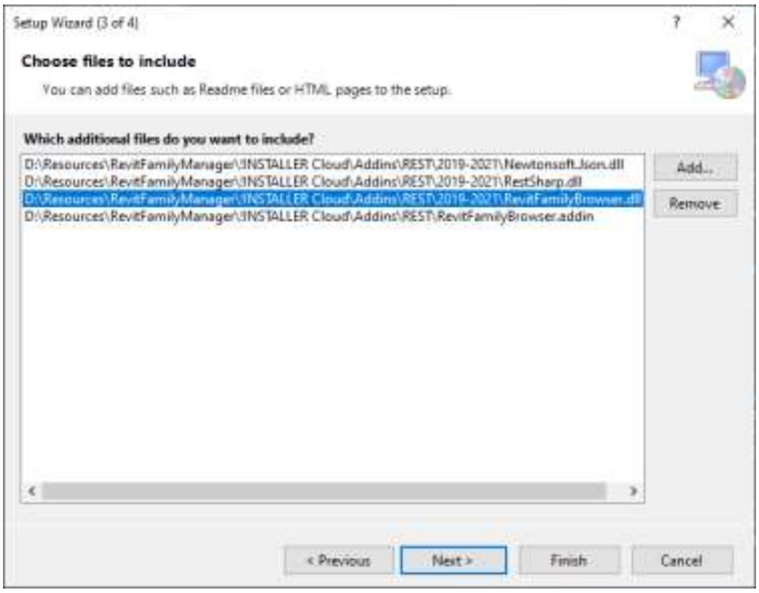
Do you want to create a redistributable package?

○ Create a merge module for Windows Installer

○ Create a downloadable CAB file

Fig. 3.26 Setup Wizard for installer

Then select the files that will be coming out of your application (fig. 3.27) via “Add” button. Appears default Window’s “Select File Dialog”, which makes files search easy and comfortable. Select Primary for all your exes, xmls, jsons, dlls, imgs etc., and then select next:


Setup Wizard (3 of 4)

Choose files to include

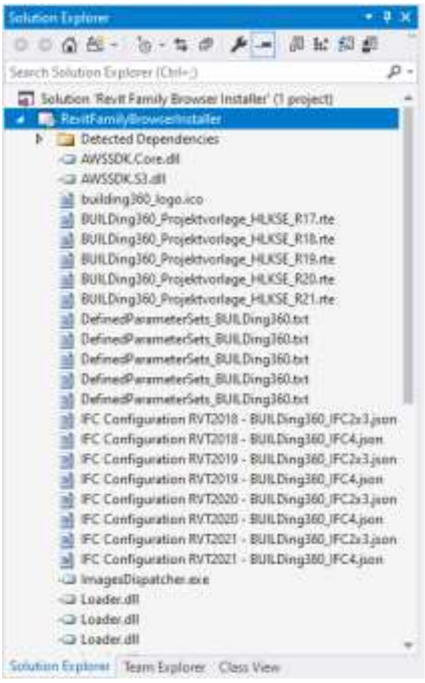
You can add files such as Readme files or HTML pages to the setup.

Which additional files do you want to include?

```
D:\Resources\RevitFamilyManager\!INSTALLER Cloud\Addins\REST\2019-2021\Newtonsoft.Json.dll
D:\Resources\RevitFamilyManager\!INSTALLER Cloud\Addins\REST\2019-2021\RestSharp.dll
D:\Resources\RevitFamilyManager\!INSTALLER Cloud\Addins\REST\2019-2021\RevitFamilyBrowser.dll
D:\Resources\RevitFamilyManager\!INSTALLER Cloud\Addins\REST\RevitFamilyBrowser.addin
```

Fig. 3.27. Setup Wizard files to include menu

After clicking finish you will see that a new project gets added to your application (fig 3.28):



Solution Explorer

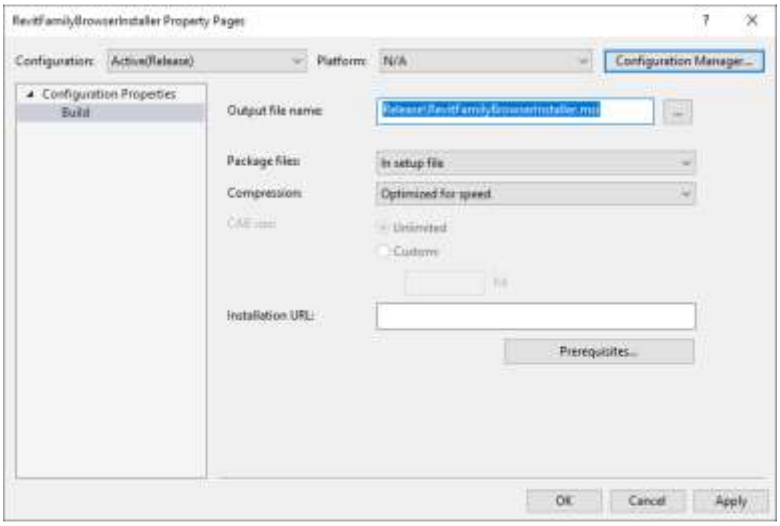
Search Solution Explorer (Ctrl+;)

Solution 'Revit Family Browser Installer' (1 project)
- RevitFamilyBrowserInstaller
  - Detected Dependencies
  - AWSSDK.Core.dll
  - AWSSDK.S3.dll
  - building360_logo.ico
  - BUILDing360_Projektvorlage_HLKSE_R17.rte
  - BUILDing360_Projektvorlage_HLKSE_R18.rte
  - BUILDing360_Projektvorlage_HLKSE_R19.rte
  - BUILDing360_Projektvorlage_HLKSE_R20.rte
  - BUILDing360_Projektvorlage_HLKSE_R21.rte
  - DefinedParameterSets_BUILDing360.txt
  - DefinedParameterSets_BUILDing360.txt
  - DefinedParameterSets_BUILDing360.txt
  - DefinedParameterSets_BUILDing360.txt
  - DefinedParameterSets_BUILDing360.txt
  - IFC Configuration RVT2018 - BUILDing360_IFC2x3.json
  - IFC Configuration RVT2018 - BUILDing360_IFC4.json
  - IFC Configuration RVT2019 - BUILDing360_IFC2x3.json
  - IFC Configuration RVT2019 - BUILDing360_IFC4.json
  - IFC Configuration RVT2020 - BUILDing360_IFC2x3.json
  - IFC Configuration RVT2020 - BUILDing360_IFC4.json
  - IFC Configuration RVT2021 - BUILDing360_IFC2x3.json
  - IFC Configuration RVT2021 - BUILDing360_IFC4.json
  - ImagesDispatcher.exe
  - Loader.dll
  - Loader.dll
  - Loader.dll

Fig. 3.28. Solution explorer for created installer project

3.8.2 Editing your installer project

To edit your project properties (fig. 3.29), for example, to change its name, right click on the project and select properties:



RevitFamilyBrowserInstaller Property Pages

Configuration: Active(Release)   Platform: N/A

Configuration Properties
- Build

Output file name: Release\RevitFamilyBrowserInstaller.msi

Package files: In setup file

Compression: Optimized for speed

CAB size: ● Unlimited ○ Custom

Installation URL:

Fig. 3.29. Editing installer project properties

There are three principle organizers in the UI (fig. 3.30): Application Folder, User's Desktop and Users Programs Menu. The Application Folder will indicate where the product is found. For instance, in the event that you wish this to contain the executable of your application, at that point duplicate the exe record, right-click the Application Folder and glue the application to it. You can likewise right-tap the executable document to make alternate ways, which can be embedded into the User's Desktop/User's Programs Menu. This empowers the alternate route to be made on the client's work area and in the fundamental Programs menu. After creation required pecking order of registries with required documents in them you get the accompanying outcome. Here are shown required ImageDispatcher.exe in Program Files index and all vital .dll/.addin records set by Revit API documentation:

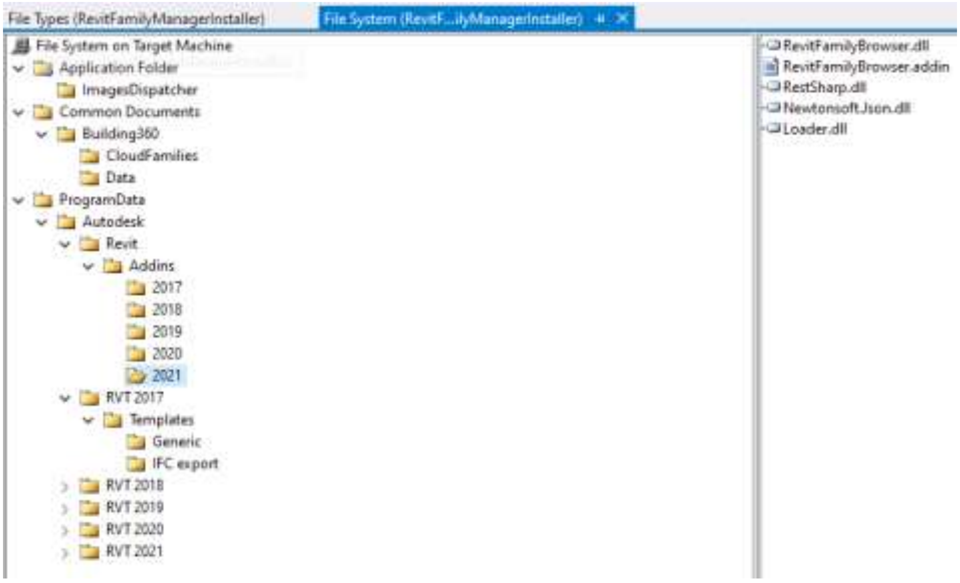


Fig. 3.30. Setting up of file system in installer project

3.8.3 Select the user’s installation folders

To change the destination of the application folder (fig. 3.31) right-click the Application Folder in the File System window and choose Properties:

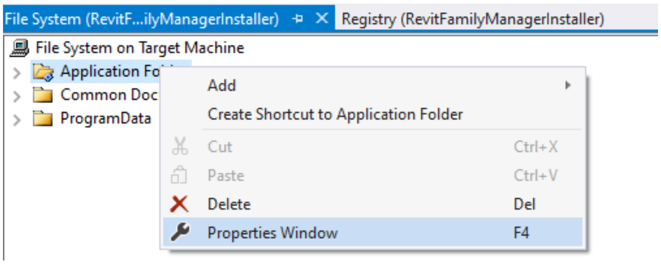
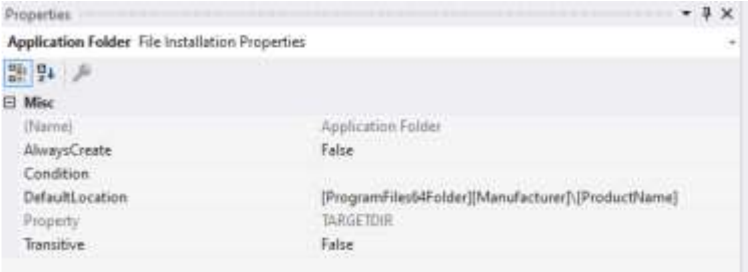

File System (RevitF...ilyManagerInstaller)
Registry (RevitFamilyManagerInstaller)
File System on Target Machine
Application Fo
Common Doc
ProgramData
Add
Create Shortcut to Application Folder
Cut Ctrl+X
Paste Ctrl+V
Delete Del
Properties Window F4

Fig. 3.31. Changing destination of the application folder

DefaultLocation (fig. 3.32) gives you the general structure of the envelope area. You may wish to eliminate the 'Producer' segment, for instance, on the off chance that you don't wish to remember this for your general envelope area. You can likewise independently adjust each of these, for example, "ProgramFilesFolder" in the event that you favor your projects to be introduced somewhere else:



Properties

Application Folder File Installation Properties

Misc

(Name)	Application Folder
AlwaysCreate	False
Condition	
DefaultLocation	[ProgramFiles64Folder][Manufacturer]\[ProductName]
Property	TARGETDIR
Transitive	False

Fig. 3.32 DefaultLocation application folder property

To create other default directories (fig. 3.33) or custom right click on “File System on Target Machine” and select “Add special folder”. The result is shown below:

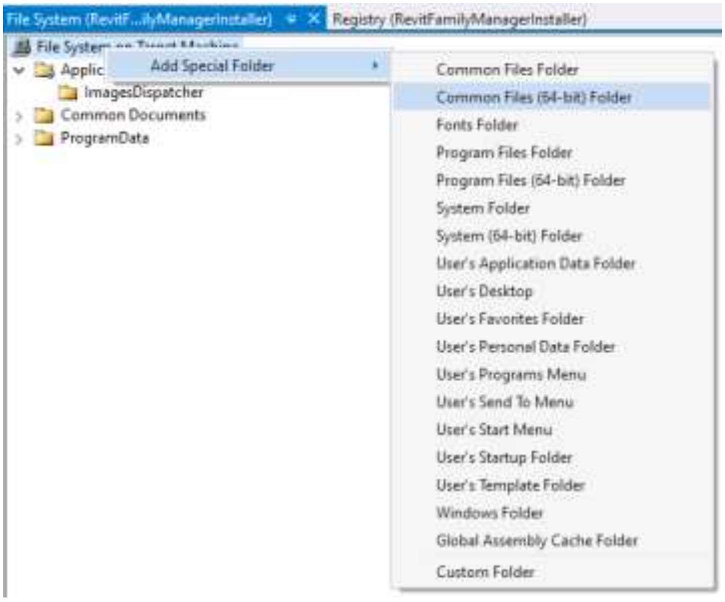
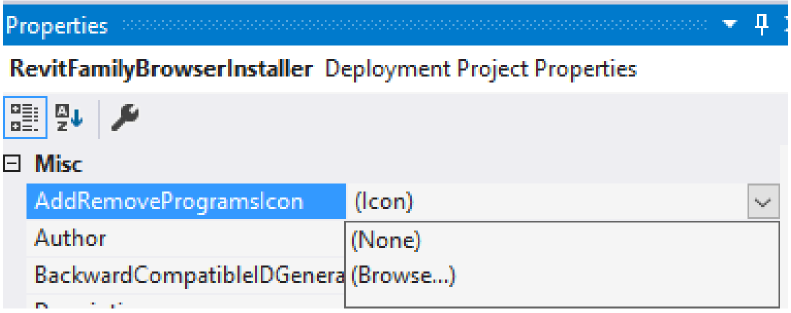


Fig. 3.33 Special folder addition process

3.8.4 Setting the application icons

In the File System menu, go to where an alternate route is found (eg User's Desktop/User's Programs Menu), right snap and pick its properties (fig. 3.34). In the properties, select the Icon field and in the drop down menu that shows up, utilize the peruse to recognize the symbol record that you need to utilize:



Properties

RevitFamilyBrowserInstaller Deployment Project Properties

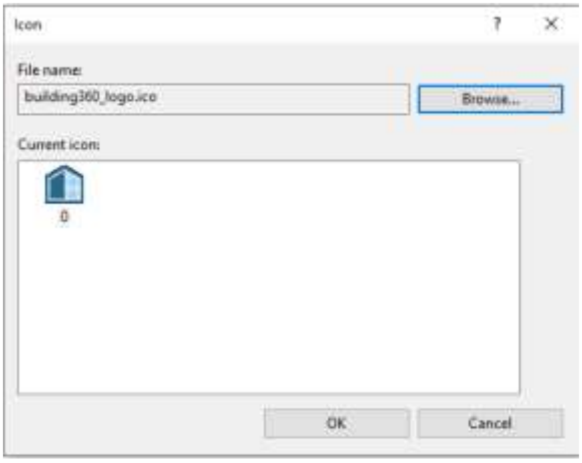
Misc

AddRemoveProgramsIcon	(Icon)
Author	(None)
BackwardCompatibleIDGenera	(Browse...)

Fig. 3.34 Adding icon for installer project

In the Icon dialog that appears use the browse button to locate the icon file (*.ico) of your choice:

In this example for the User’s Desktop I selected ‘Add File…’ and used the icon I had developed for the “RevitFamilyBrowser” (fig. 3.35) project I was working on, located in the ‘res’ folder of the said project:



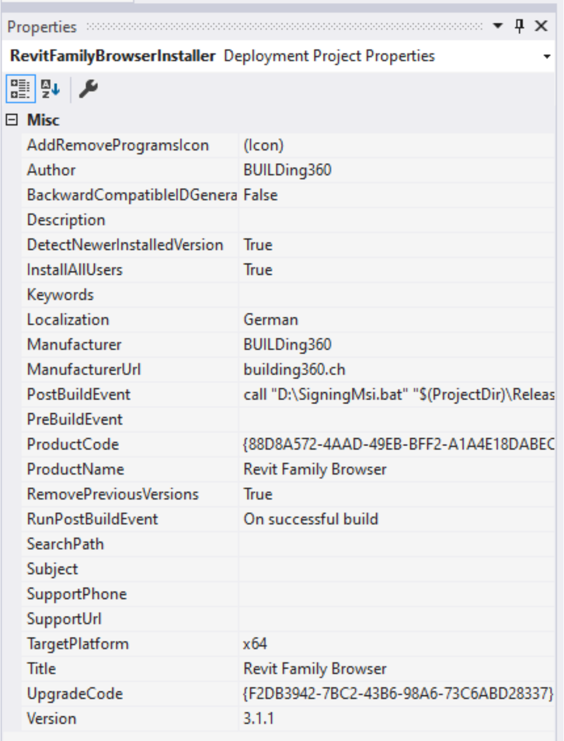
File name: building360_logo.ico

Current icon:

Fig. 3.35 Selected FamilyBrowser icon

3.8.5 Modifying all needed properties of the installer project

It is conceivable to change program's adaptation, which will be shown in Programs and Features control board, introducing alternatives like "InstallAllUsers" or "DetectNewerInstalledVersion", setting up Author/Manufacturer choices, evolving Post/PreBuildEvents and so forth (fig 3.36):

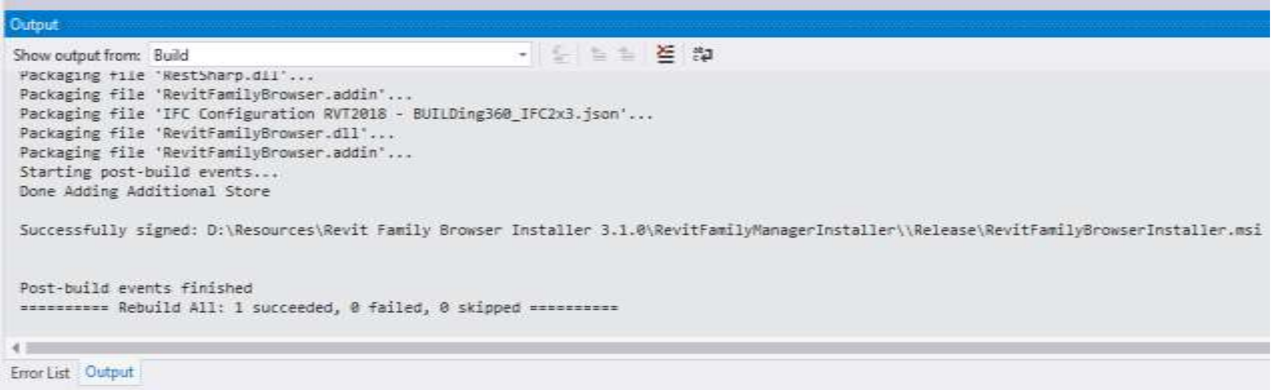


AddRemoveProgramsIcon	(Icon)
Author	BUILDing360
BackwardCompatibleIDGenera	False
Description	
DetectNewerInstalledVersion	True
InstallAllUsers	True
Keywords	
Localization	German
Manufacturer	BUILDing360
ManufacturerUrl	building360.ch
PostBuildEvent	call "D:\SigningMsi.bat" "$(ProjectDir)\Releas
PreBuildEvent	
ProductCode	{88D8A572-4AAD-49EB-BFF2-A1A4E18DABEC
ProductName	Revit Family Browser
RemovePreviousVersions	True
RunPostBuildEvent	On successful build
SearchPath	
Subject	
SupportPhone	
SupportUrl	
TargetPlatform	x64
Title	Revit Family Browser
UpgradeCode	{F2DB3942-7BC2-43B6-98A6-73C6ABD28337}
Version	3.1.1

Fig 3.36. Installer project main properties to modify

3.8.6 Building your installer project

By default when you build your project it will build your main application. Build it by right-clicking on the installer project and choosing Build you will receive the following output in “Output” window (fig. 3.37):



```
Packaging file 'RestSharp.dll'...
Packaging file 'RevitFamilyBrowser.addin'...
Packaging file 'IFC Configuration RVT2018 - BUILDing360_IFC2x3.json'...
Packaging file 'RevitFamilyBrowser.dll'...
Packaging file 'RevitFamilyBrowser.addin'...
Starting post-build events...
Done Adding Additional Store

Successfully signed: D:\Resources\Revit Family Browser Installer 3.1.0\RevitFamilyManagerInstaller\\Release\RevitFamilyBrowserInstaller.msi


Post-build events finished
========== Rebuild All: 1 succeeded, 0 failed, 0 skipped ==========
```

Fig. 3.37. Output of building installer project

Conclusions

During this part there was demonstrated and described how to create entire infrastructure of Revit plug-ins along with Amazon Web Services to store data, files, send notifications etc. for modeling software of engineering equipment, in this case Autodesk Revit via Revit API for .NET. There was particularly explained working principles and functions of Revit Family Browser plug-in and its standards-controlling ecosystem.

The REST service built with the help of Laravel framework was described step by step and its features have been displayed, demonstrating its flexibility, configuration and simpleness.

This part showed how to create installer project and the entire process of creation .msi package for easy distribution of application through Windows computes.

Revit Family Browser application infrastructure with plug-ins and Amazon cloud services stands out with the following advantages:

– increases designing efficiency and speed;

– rich functionality with multiple settings;

– strong standards for projects

– simple new families upload pipeline

– family versions switch

– multilingual support and locations settings for content

– statistics and global search feature to show the most popular items

– REST service for universal and quick data retrieval and uploading

– drag-n-drop types to insert;

– user-friendly design;

– no more loading families via "Load Family" and searching through numerous folders within windows explorer and loading a project with multiple redundant types;

– flexible categories and families management;

– moveable/collapsible browser always stays open and accessible;

– GUI for creating/modifying type catalogs;

– filters to ease navigation through types;

– silent deployment options to firm wide user base;

– available for Autodesk Revit 2017, 2018, 2019, 2020, 2021.

CONCLUSIONS

Building Information Modeling software solutions today foster 3D intelligent modeling beyond simple geometric forms and shapes. Best practices in BIM implementation have demonstrated the ease of prototyping for designs with non-typical complexities and magnitudes.

BIM is a moderately new innovation particularly in the development area, an industry commonly delayed in adjusting to changes. BIM advocates guarantee that soon, it will offer a ton of significant worth as far as:

- improving perception;
- improving profitability through simple data recovery;
- increasing coordination of development records;
- linking of crucial data, for example, merchants for explicit materials, the area of subtleties and amounts needed for offering;
- increasing pace of conveyance;
- reducing generally speaking expenses.

Building data demonstrating and mechanized amounts innovations can furnish the business with weighty occasions to raise the nature of the business to a lot higher and refined level. Having the ability to reproduce a scope of information choices with continuous cost guidance and carry on all through the nitty gritty plan, development, and operational stages, BIM will without a doubt put development rehearses at a higher worth.

Critical piece of BIM is Computer-aided Design. Computer aided design drawings offer the adaptability to draft and plan in a computerized circle, which were recently done by hand. The computerized design makes information dealing with simpler, more secure, and snappier. Earlier hand drawn diagrams can be checked and afterward can be developed carefully. Numerous CAD programs are presently utilizing three-dimensional drawings to boost efficiency and give speedier, better item results, taking into consideration the advancement of the smallest subtleties.

In present day displaying frameworks the capacity to work with programming systems is set up at the center level. PC designs just need to pick which programming advancement framework will be utilized to make tasks to control BIM objects.

To understand the problem of current research itself there was implemented numerous experiments including real engineers and architectures. The experiment was based on the time consumed during similar projects and the average amount of mistakes made with implemented tool and without. Moreover, the feedback of those specialists has been accounted to make a final product more comfortable and well-to-do not only from the sight of the computer engineer but from the sight of BIM specialists.

For the implementation of the practical part of my graduation research an integrated development environment Visual Studio and C# programming language was chosen for the creation of the system because of the exhaustive documentation and flexibility of this framework.

In my diploma work there was shown how to create Revit plug-in for manipulation of modeling engineering equipment computer system.

The developed plugin Revit Family Browser has the following advantages: increases efficiency, drag-n-drop feature for families to load/insert, no need for loading families via numerous buttons and searching through folders within windows explorer and loading a project with multiple unwanted types, moveable/collapsible browser always open and accessible, GUI for creating/modifying type catalogs, silent deployment option and it is available for all common versions of Autodesk Revit.

As a result of research through surveys of current BIM engineers and specialists, the following product is highly recommended to engineers of any direction (Electrics / Heating / Ventilation / Air conditioning), as well as to architects and BIM coordinators, because implemented infrastructure greatly simplifies the design and compliance with standards. The evidence of this is the successful practical application of this product by many companies in Ukraine, Switzerland, Germany, and some other countries. Thanks to a wide user base and the feedback mechanism, the entire infrastructure is actively improving and evolving over time.

REFERENCES

1. What it BIM? [Internet Resources] – marketing.aco. Access mode: https://www.marketing.aco/bim/what-is-bim.

2. Rowland, S. BIM to IoT: the persistence problem. Lecture Notes of the Institute for Computer Sciences / Rowland, S : Springer – Porto: 2016 – 138.

3. Rudden, K. BIM and ISO 19650 from a project management perspective / Rudden, K. – Booklet on ISO Standard / Efca. Bern – 48.

4. Sawhney, A. Building Information Modelling for Project Managers / Sawhney, A., Khanzode, A., Tiwari S. : RISC. London – 42.

5. Bazjanac, V. IFC BIM-Based Methodology for Semi-Automate Building Enerfy Performance Simulation / Bazjanac V. // Berkeley – 13.

6. Ziolkowski, J., Bartyzel, M. How Revit API can boost application of company and project-specific workflows / Autodesk University. London – 50.

7. Preton, S. AutoCAD .NET Developer's Guide / Preton, S. // Intellectual Property. – 541.

8. Eastman, C. M. BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors / Eastman C. M. – John Wiley & Sons – Hoboken, NJ: 2011 – 491.

9. Carrillo, E. IoT framework for smart buildings with cloud computing. IEEE First International Smart Cities Conference / Carrillo E., Benitez V., Mendoza C. : 2015 – 113.

10. BIM Consulting Services | Dynamo vs Revit API [Internet Resources] – elogictech.com, 2019. – Access Mode: https://www.elogictech.com/blogs/bloginfo/dynamo-vs-revit-api

11. Lesson 3: A first Look at Code [Internet Resources] – knowledge.autodesk.com, 2020. – Access mode: https://knowledge.autodesk.com/support/revit-products/learn-explore/caas/simplecontent/content/lesson-3-first-look-code.html.

12. Amazon Web Services (AWS) – Cloud Computing Services [Internet Resources]: Amazon Web Services, Inc; Access mode: https://aws.amazon.com. – Title from the screen.

13. Amazon Elastic Cloud | EC2 [Internter Resources] – aws.amazon.com, 2020 – Access mode: https://aws.amazon.com/ec2.

14. Cloud Object Storage | Amazon Simple Storage Service (S3) [Internter Resources] – aws.amazon.com, 2020 – Access mode: https://aws.amazon.com/s3.

15. AWS Identity & Access Management | IAM [Internter Resources] – aws.amazon.com, 2020 – Access mode: https://aws.amazon.com/iam/.

16. Amazon Virtual Private Cloud | VPC [Internter Resources] – aws.amazon.com, 2020 – Access mode: https://aws.amazon.com/en/vpc.

17. Tammik, J. Revit 2013 API My First Plugin Training [Internet Resources] – thebuildingcoder.typepad.com, 2013. – Access mode: https://thebuildingcoder.typepad.com/files/revit_2013_api_my_first_plug-in_training.pdf

18. Family Browser Help Page [Internet Resources]: Building360; Access mode: https://www.building360.ch/FamilyBrowser/en – Title from the screen.

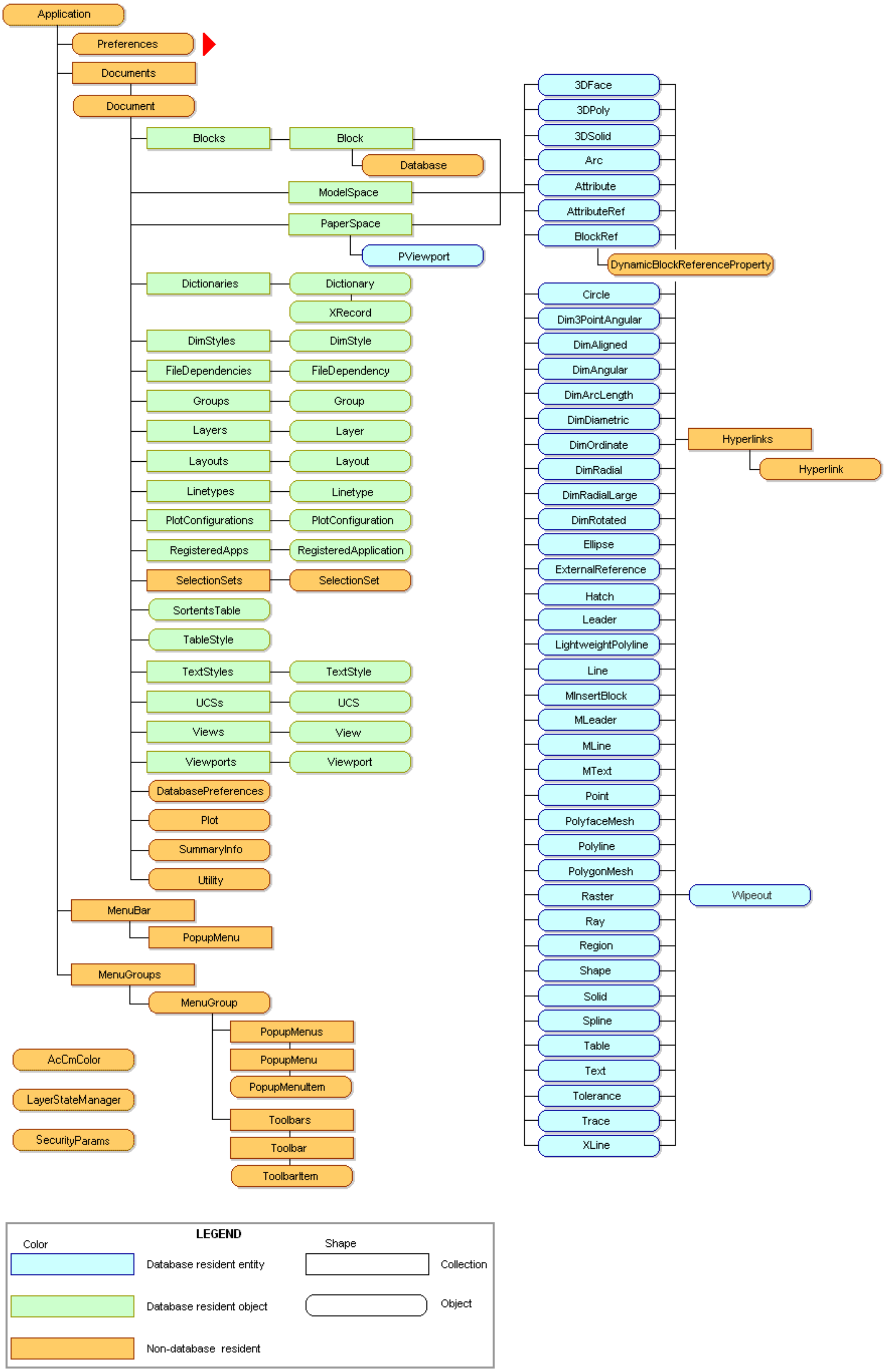
19. Guide of Revit Image Printer [Internet Resources]: Building360; Access mode: https://www.building360.ch/ImagePrinter, free – Title from the screen.

20. Castelo, A. Laravel API Tutorial: How to Build and Test a RESTful API [Internet Resources] – Toptal.com, 2017. – Access mode: https://www.toptal.com/laravel/restful-laravel-api-tutorial.

21. ДСТУ ISO/IEC 15910:2012, IDT. Документування програм. Документація користувача / Київ: «Український науково-дослідний і навчальний центр проблем стандартизації, сертифікації та якості», 2013. – 37 с.

APPENDIX A

Autodesk Object Model (ActiveX)



APPENDIX B

The main class of Revit Family Browser

```
using Autodesk.Revit.DB;
using Autodesk.Revit.UI;
using System;
using System.Collections.Generic;
using System.Linq;
using Autodesk.Revit.DB.Plumbing;
using RevitFamilyBrowser.Data;
using Autodesk.Revit.DB.Events;
namespace RevitFamilyBrowser
{
    public class SingleInstallEvent : IExternalEventHandler
    {
        public Document Document { get; set; }
        public string MEPSystemName { get; set; }
        public void Execute(UIApplication uiapp)
        {
            UIDocument uidoc = uiapp.ActiveUIDocument;
            Autodesk.Revit.ApplicationServices.Application app = uiapp.Application;
            Document doc = uidoc.Document;
            Document = doc;
            MEPSystemName = Properties.Settings.Default.FamilyPath;
            string FamilyPath = Properties.Settings.Default.FamilyPath;
            string FamilyType = Properties.Settings.Default.FamilyType;
            string FamilyName = Properties.Settings.Default.FamilyName;
            if (FamilyName.Equals("BuiltInFamily") ||
FamilyName.Equals(string.Empty))
            {
                Element element = null;
                //Collect Ducts
                FilteredElementCollector collector = new
FilteredElementCollector(doc);
                collector.OfCategory(BuiltInCategory.OST_DuctCurves);
                element = GetElementFromTemplate(collector, FamilyType);
                if (element != null)
                {
                    InsertElement(uidoc, element);
                }
            }
            else
            {
                FilteredElementCollector collector = new
FilteredElementCollector(doc).OfClass(typeof(Family));
                FamilySymbol symbol = collector.FirstElement() as FamilySymbol;
```

Part 2 of the Appendix B

```
Family family = FindFamilyByName(doc, typeof(Family), FamilyPath) as Family;
                if (family == null)
                {
                    using (var transaction = new Transaction(doc, "InsertTransaction"))
                    {
                        transaction.Start();
                        try
                        {
                            if (!doc.LoadFamily(FamilyPath, out family))
                            {
                                CommonMethods.ProcessError("warningFamilyMissed", true,
FamilyName);
                            }
                        }
                        catch (Exception exc)
                        {
                            CommonMethods.ProcessError("warningFamilyMissed", true,
FamilyName);
                        }
                        transaction.Commit();
                    }
                }
                ISet<ElementId> familySymbolId = family.GetFamilySymbolIds();
                foreach (ElementId id in familySymbolId)
                {
                    // Get name from buffer to compare
              if(CommonMethods
                  .CorrectName(family.Document.GetElement(id).Name)
                  .ToLower() ==
                  CommonMethods
                  .CorrectName(FamilyType)
                  .ToLower()
                            && FamilyType != null)
                    {
                        symbol = family.Document.GetElement(id) as FamilySymbol;
                    }
                }
                if (symbol != null)
               uidoc.PostRequestForElementTypePlacement(symbol);
            }
        }
        private Element FindFamilyByName(Document doc, Type targetType, string
familyPath)
        {
```

```
if (familyPath != null){
        int indexSlash = familyPath.LastIndexOf("\\") + 1;
        string FamilyName = familyPath.Substring(indexSlash);
        string targetName = FamilyName.Substring(0, FamilyName.Length - 4);
        return
          new FilteredElementCollector(doc).OfClass(targetType)
            .FirstOrDefault(e => e.Name.Equals(targetName));
      }
         return null;
    }
    private Level GetCurrentLevel(Document doc)
    {
      Level currentLevel = null;
      FilteredElementCollector levelCollector = new
FilteredElementCollector(doc);
      levelCollector.OfClass(typeof(Level)).ToElements();
      foreach (Element lvl in levelCollector)
      {
        currentLevel = lvl as Level;
      }

      return currentLevel; }
    private void InsertElement(UIDocument uidoc, Element element)
    {
      Document doc = uidoc.Document;
      FilteredElementCollector elementCollector = new
FilteredElementCollector(doc);

      BuiltInCategory category =
(BuiltInCategory)Enum.Parse(typeof(BuiltInCategory),
element.Category.Id.ToString());
      elementCollector.OfCategory(category).OfClass(typeof(ElementType));
      foreach (ElementType item in elementCollector)
       {
                  if (item.Name.Equals(element.Name))
                  {
                       ElementType elementType = item as ElementType;

       uidoc.PostRequestForElementTypePlacement(elementType);
                  }
       }
    }

    private Element GetElementFromTemplate(FilteredElementCollector
collector, string elementName)
```

```
        {
```

End of the Appendix B

```
               Element element = null;
            foreach (Element item in collector)
            {
                if (item.Name.Equals(elementName))
                {
                    element = item;
                }
            }
            return element;
        }
    }
}
```