Dynamic Blocks - Course Overview

Day 1

Actions 101 – Dynamic Blocks
Focus: Conceptual Understanding Over Step-by-Step Exercises

This intensive first day introduces all core Action types used in Dynamic Blocks and explains how they work together to control geometry. While not every exercise will be completed, every concept will be explained so participants understand the logic behind building flexible, intelligent blocks.

The emphasis is on understanding relationships:

- Parameter → Action → Geometry → Behavior

By the end of the day, learners will understand how to design, combine, and troubleshoot dynamic block actions.

Learning Objectives

Participants will:

- Understand how Actions control geometry
- Learn all major Action types and when to use them
- Understand how multiple actions can work from a single parameter
- Learn how to control movement direction, scaling, and rotation
- Understand chained vs unchained behavior
- Recognize and fix common problems
- Gain the confidence to experiment and build independently

Course Structure

1) Foundations: How Actions Work

Core Concept

- Parameters define *how* a block can change
- Actions define *what happens* when the parameter is used
- Geometry responds based on selection sets

Key Principles

- A parameter without an action does nothing
- Actions only affect selected geometry
- Grip points control user interaction

---

2) Core Action Types (Fundamentals)

These are the building blocks of nearly all dynamic blocks.

Stretch

- Change length or position of geometry
- Works with Linear parameters
- Selection windows determine what stretches vs moves

Move

- Reposition geometry
- Often paired with Point or Linear parameters
- Used for relocating components

Rotate

- Spin geometry around a defined base point
- Driven by Rotation parameters
- Useful for pivoting parts

Scale

- Resize geometry proportionally
- Controlled by Linear parameters
- Parameter placement affects scale direction

Flip

- Mirror geometry across an axis
- Controlled by Flip parameters
- Selection sets determine what flips

Visibility States

- Show/hide geometry variations
- Allows multiple versions of a block in one

---

3) Controlling Behavior with Properties

Key properties that dramatically change how actions behave:

- Distance Multiplier
    - Makes objects move proportionally
    - Maintains relative positioning
- Angle Offset
    - Changes direction of movement
    - Enables multi-direction stretch from one parameter
- Grip Count
    - Controls user interaction points

These settings allow one parameter to drive multiple outcomes.

---

4) Combining Actions on One Parameter

Core concept: One grip can control many elements.

Examples explored conceptually:

- Stretching multiple parts from one dimension
- Moving objects at different rates
- Maintaining layout relationships
- Keeping objects centered while resizing

This is where blocks begin to feel “intelligent.”

---

5) Advanced Stretch Concepts

Understanding how to manipulate geometry in multiple directions:

- Multi-direction stretch
- Using Angle Offset to redirect movement

- Using Distance Multiplier to maintain ratios
- Proportional resizing from a single control

This introduces controlled complexity without adding extra parameters.

---

6) Array Actions

Using parameters to generate repeating geometry:

- Auto-growing patterns
- Incremental spacing
- Minimum and maximum limits
- Combining Stretch + Array for scalable designs

Ideal for:

- Holes
- Panels
- Modular components

---

7) Polar Actions

Combining movement types:

- Rotate + Stretch at the same time
- Radial control from one grip
- Directional expansion

Used for:

- Radial layouts
- Hinged components
- Adjustable arms

---

8) Unchained Parameters

Key concept: independence.

- One parameter does NOT trigger another

- Objects can move separately
- Enables complex mechanical behavior

Example ideas:

- Rotating one part without affecting the rest
- Independent movement zones

---

9) Chained Actions

Key concept: linked behavior.

- One action triggers another
- Movement cascades through the block
- Used to keep elements aligned

Applications:

- Flipping without mirroring certain parts
- Keeping text readable
- Maintaining relative positions

Understanding when to use and when to avoid chaining is critical.

---

10) Moveable Base Point

Advanced usability concept:

- The insertion point can shift
- Base point can follow geometry
- Improves placement flexibility

Useful for:

- Orbiting objects
- Components with changing centers
- Alignment-driven blocks

---

11) Multi-Parameter Systems

How complex blocks are built:

- Multiple parameters working together
- One parameter controlling geometry AND other parameters
- Layered behavior

This is the bridge between simple and professional-level dynamic blocks.

---

12) Problem Solving & Troubleshooting

Common issues covered:

- Geometry stretching the wrong way
- Grip points missing
- Geometry moving twice
- Flip behaving incorrectly
- Parameters not moving with geometry
- Strange behavior from constraints

Key mindset:

- Selection sets control behavior
- Action properties control direction and scale
- Sometimes rebuilding is faster than fixing

Teaching Strategy for the Day

- Demonstration-led
- Concept explanation first
- Selected exercises used to illustrate ideas
- Not every step completed live
- Emphasis on understanding logic

Expected Outcomes

By the end of the day, participants should:

- Understand every major Action type
- Know how to attach actions to parameters
- Understand how to combine multiple actions
- Be able to predict how geometry will respond
- Recognize common setup mistakes
- Feel confident experimenting independently

Key Takeaway Message

You don’t need to memorise steps.

If you understand:

- What the parameter controls
- What the action affects
- What geometry is selected

You can build almost any dynamic block.

Homework – Practice Exercises (Actions 101)

To reinforce today’s learning, complete the practice exercises provided in the course materials. The goal is not perfection, but exploration and understanding how actions interact with parameters and geometry.

Your Task

Work through the following practice activities at your own pace:

- Practice What You’ve Learnt
- Advanced Stretch practice
- Polar Stretch practice
- Unchained Parameters (Parts 1 & 2)
- Chained Parameters
- The Crane Challenge (optional stretch task)

Focus Areas

While completing the exercises, pay attention to:

- How one parameter can control multiple actions
- How selection sets affect behavior
- The effect of Distance Multiplier and Angle Offset
- Differences between chained and unchained actions
- Keeping geometry aligned and proportional

Try This

As you work, experiment by:

- Changing multiplier values
- Adjusting angle offsets
- Adding or removing geometry from selection sets
- Turning chained actions on and off

Observe what changes, this is where the real learning happens.

Optional Challenge

Choose one exercise and try to:

- Solve it using a different action type
- Simplify it using fewer parameters
- Improve how the block behaves

Bring to Next Session

Come prepared to:

- Show one exercise you found difficult
- Share one discovery or “aha” moment
- Ask questions about anything that didn’t behave as expected

Remember: mistakes are part of the process, they usually teach more than getting it right first time.

Day 2

Parameters & Constraints 101 – Dynamic Blocks
Focus: Understanding Constraint Logic & Parametric Control

Course Overview

This intensive course introduces the full set of core concepts behind geometric and dimensional constraints in Dynamic Blocks. The emphasis is on understanding how constraints control shape, size, position, and relationships between objects.

Not every step of every exercise will be completed in class. Instead, the course focuses on understanding the principles that allow you to build intelligent, flexible, and predictable parametric geometry.

By the end of the session, participants will understand how to fully constrain geometry, create dependent relationships, and build blocks that adapt intelligently to change.

Learning Objectives

Participants will:

- Understand how geometric constraints control shape
- Understand how dimensional constraints control size
- Learn how to build fully constrained, stable geometry
- Understand the difference between rotatable and fixed geometry
- Learn to create dependent parameters using formulas
- Explore arrays, user parameters, and block tables
- Gain confidence troubleshooting constraint conflicts

Course Structure

1) Foundations: What Are Constraints?

Core Concept

- Constraints control how geometry behaves when edited
- Geometric constraints control relationships
- Dimensional constraints control size and distance
- A fully constrained object behaves predictably

Key Principles

- Always start with a Fix constraint (your zero point)
- Build relationships before adding dimensions
- Choose constraint types based on intended movement

---

2) Geometric Constraint Fundamentals

Introduction to the most important constraint types:

- Horizontal / Vertical
- Parallel / Perpendicular
- Coincident
- Tangent
- Equal
- Fix

Focus:

- Preventing unwanted rotation
- Maintaining shape integrity
- Keeping geometry connected

---

3) Fully Constraining Basic Geometry

Using simple shapes to understand stability:

Rectangle Concepts

- Two approaches:
    - Non-rotatable geometry
    - Rotatable geometry
- When to use Linear vs Aligned dimensions
- Understanding constraint direction and grip placement

Key takeaway:
How constraint choices affect future flexibility.

(Concept reference:)

---

4) Constraining Complex Shapes

Building relationships between lines and curves:

Oblong Concepts

- Maintaining smooth transitions
- Using Tangent constraints
- Making multiple elements equal
- Controlling radius with dimensional constraints

Key takeaway:
How to preserve shape while allowing resizing.

(Concept reference:)

---

5) Angular Control & Parameter Management

Working with non-90° geometry:

- Angular dimensional constraints
- Equal constraints across multiple sides
- Coincidence for connection control
- Introduction to the Parameters Manager
- Hiding constraints from the end user

Key takeaway:
How to manage complex parametric systems cleanly.

(Concept reference:)

---

6) Creating Rotatable Geometry

Building objects that resize AND rotate:

- Concentric constraints
- Tangent relationships
- Parallel + Perpendicular logic
- Angular parameters for rotation control

Key takeaway:
How to allow movement while preserving structure.

(Concept reference:)

---

7) Central-Point Design Thinking

Working from a fixed reference:

Clock Exercise Concepts

- Building from a central point
- Independent rotation using angular constraints
- Maintaining alignment while rotating components

Key takeaway:
Designing from a logical origin point improves control.

(Concept reference:)

---

8) Symmetry, Coincidence & Structural Relationships

Building connected systems:

- Symmetric constraint for mirrored behaviour
- Coincidence for linking parts
- Collinear relationships
- Using Equal constraints for consistency

Key takeaway:
How to keep parts moving together naturally.

(Concept reference:)

---

9) Applied Constraint Projects

Using multiple techniques together:

Crane Exercise Concepts

- Connecting separate components
- Creating moving assemblies
- Combining angular and linear parameters
- Controlling dependent movement

Key takeaway:
How multiple constraints form a working mechanism.

(Concept reference:)

---

10) Automatic Constraints & Editing

Understanding system-generated relationships:

- Using CONSTRAINTINFER
- Reviewing automatic constraints
- Editing and removing unwanted constraints
- Recognizing over-constrained geometry

Key takeaway:
Constraints must be intentional, not accidental.

(Concept reference:)

---

11) Parametric Logic with Formulas

Creating intelligent relationships:

- Naming dimensional constraints
- Making dimensions dependent on others
- Centering objects with formulas
- Maintaining proportions automatically

Examples:

- Width = 2 × Height
- Position = Width ÷ 2

Key takeaway:
Formulas turn static geometry into smart systems.

(Concept reference:)

---

12) Arrays & User Parameters

Scaling geometry intelligently:

- Using formulas to control array count
- User-defined parameters
- Driving layout with dimensions
- Incremental resizing behaviour

Key takeaway:
Controlling repetition through logic.

(Concept reference:)

---

13) Block Tables (Advanced)

Creating preset configurations:

- Linking parameters to table values
- Driving geometry from predefined options
- Using arrays with parameter tables

Key takeaway:
Turning parametric blocks into configurable tools.

(Concept reference:)

Teaching Strategy

- Concept-led demonstrations

- Selected exercises used to illustrate principles
- Not every step completed live
- Focus on logic rather than memorisation

Expected Outcomes

By the end of the day, participants should:

- Understand how to fully constrain geometry
- Know how to prevent unwanted movement
- Be able to create resizable and rotatable shapes
- Understand how constraints interact
- Use formulas to control design logic
- Recognize and fix constraint conflicts

Key Takeaway Message

Constraints are about control and predictability.

If you understand:

- Where to fix geometry
- How to build relationships
- When to use dimensional vs geometric constraints

You can create stable, intelligent parametric blocks that behave exactly as intended.