PH7X SYSTEMS ®

0.1.0 · CHAPTER 33

Size each wave to the people who accept it, and write down what stops the next

SharePoint Migration Field Guide · Chapter 33

pH7x Systems

2026-11-07

33. Size each wave to the people who accept it, and write down what stops the next

> The platform decides how fast content moves. The people who must say it arrived decide how fast the project does.

Chapter 29 gave every group of sites a route, after Chapters 14, 26 and 27 settled which sites go at all. Chapter 9 runs one wave of a copy, and Chapter 23 attaches one database. This chapter decides what moves together, in what order, and what halts the rest when a wave fails.

> **In what order does the estate move, and what stops the next wave?**

The decision this chapter leads to is the wave calendar: each wave with its sites, its owners and validators, its window and its stop criterion, written before the first wave starts.

Which sites this chapter is for

Every site collection whose disposition is to migrate, copied into SharePoint Online (Chapters 8 and 9) or attached to Subscription Edition. The pilot and its rehearsal, Chapter 31, come first and time what this chapter plans with, so a calendar written before them is a draft. Microsoft's guide says as much: "Assess the data from the pilot migration to determine the rest of your migration schedule" (team sites migration guide, checked on 7 November 2026).

What sizes a wave

A wave is the sites that freeze, move, are verified and cut over together: the unit a stop acts on. Six things bound it, and the last usually decides.

WHAT SIZES A WAVE	COPY INTO SHAREPOINT ONLINE	ATTACH TO SUBSCRIPTION EDITION	HOW TO MEASURE IT
The unit that moves	A site collection, in a task of its own	A content database, with every site collection in it	Chapter 2's inventory
What runs in parallel	Tasks against different site collections	After the first is verified, attaches started minutes apart	The rehearsal's logs
What sets the pace	Throttling, tighter on weekday daytime; your own rate	The database server and its storage	Timed on your own content
What users lose meanwhile	Writes, during the final delta and verification	Writes, from each database's backup to its cutover	The freeze each owner accepts, in writing
What cuts over together	One wave, or the whole estate at one event	Every database behind one address	The dependencies below
Who accepts it	The wave's validators, in hours they confirmed	The same	Hours per kind of site, timed on the pilot

Sources for the table: migration performance guidance, copy databases and upgrade content databases, all checked on 7 November 2026.

A copy: parallel by site collection, paced by the tenant

Microsoft encourages "parallel tasks against different site collections, if possible", recommends no more than 5,000 migration jobs in the queue, and names "migrating all your content in a single task or attempting to migrate during peak hours" among the causes of throttling (migration performance guidance). Chapter 29 holds the rest: throttling tighter in weekday daytime hours, daily maxima that are ceilings, a rate measured on your own content, Microsoft's tool on several computers at once, and every route in one tenant as one load.

So a copy wave has two sizes. Its initial passes run in the background, waves ahead if need be. Its freeze is what people feel, and the final delta, the verification read and the validators' acceptance have to fit inside it (Chapter 9). Size the freeze from the measured drift and rate, and the wave from the freeze its owners accept.

An attach: one database first, the rest minutes apart

Microsoft's sequence starts with one database: restore it and verify the restore "before you restore the other databases" (copy databases), attach it, and continue only "after you restore the first content database and verify success". The others attach in parallel from separate PowerShell sessions started "several minutes" apart because of "temporary locks set for the web application" (upgrade content databases). That page sets no maximum. Microsoft's page on installing updates, whose database upgrade phase it calls "essentially the same" as a version upgrade's, recommends no more than two content databases upgrading at once on one SQL Server volume, and typically four per web or

application server (install a software update, checked on 7 November 2026). Whether those figures hold for a version upgrade we do not know; we would start the rehearsal there and keep what it measures.

The window is the read-only period. Microsoft's overview sets the old farm read-only "so that users can continue to access the old farm while upgrade is in progress", and copies the content and service application databases in one stage (upgrade overview, checked on 7 November 2026); nobody adds new site collections "until you have restored all the content databases" (upgrade content databases). We found no Microsoft page that spreads one farm's attach over several windows. A calendar that does is your design, with four consequences: each database stays read-only from its backup to its cutover; a term, profile or stored credential changed on the old farm after the service databases were copied stays there unless re-entered; search on the new farm waits for a full crawl after the last content database (Chapter 11); and every database behind one address cuts over with it, since each web application keeps its address (Chapter 23) and we know of no documented way to serve one address from two farms.

So an attach wave is a set of addresses with every database behind them: for path-based sites, a whole web application. When its validators cannot accept it inside the read-only window, the window grows, with its owners' agreement, or more validators are named; a web application too large for any window is a structure question for Chapter 12.

Group by what must move together

Draw waves from dependencies first and convenience last:

- **Web applications and their databases**, for an attach. A site collection travels with its database; regrouping means `Move-SPSite` on the old farm, before the backup, under the conditions Chapter 26 gives.
- **Sites and systems that depend on each other**: an integration whose other end re-points once (Chapter 22), a workflow or search configuration that spans sites, sites whose pages and navigation link into each other. They share a wave, or the dependency gets a written interim state, a redirect or a link to the old address, with an owner and an end date.
- **Owners and validators.** Chapter 13's administrator list is the first draft; a team whose sites share a wave switches once.
- **The business calendar.** No site moves during the peak of the cycle Chapter 14 recorded for it: a year end, an audit, a close.

Then the order: the pilot; whatever others depend on before what depends on it, on an attach the root site collection's database and the My Site host's first (Chapter 23). We would make the first wave representative and well staffed rather than the most critical, because the first wave sets the precedent (Chapter 6). A site with no validator joins no wave until the owner of Chapter 6's ownership rule names one: Chapter 13 found such sites hold the project open at the end.

Size each wave to the people who accept it

Two checks close a wave. Chapter 8's comparison belongs to the migration team and scales with machines. Acceptance belongs to the validators Chapter 1 named, who have day jobs and whose availability, it said, is usually the real constraint. Size each wave to it, person by person:

1. On the pilot, time each validator's acceptance of each kind of site: a library site, a site of lists and forms, an application with workflows.
2. For each candidate wave, list every validator's sites and the hours they take.
3. Ask each validator, and whoever sets their work, for the hours they will give in that window, in writing.
4. Give no validator more sites than those hours cover, and take the previous wave's hypercare (Chapter 9) out of the same hours.

Do not add the hours into a team total: a total hides the validator who holds most of the sites. **A wave is as large as its validators can accept in its window, never as large as the tool can copy in a weekend.**

Deltas across waves, and how many cutovers

For a copy, each wave runs Chapter 9's whole sequence, from its own baseline to its own verification record. Waves overlap: the next wave's baseline and initial pass run while the previous one is in its deltas, inside the same tenant load.

Microsoft recommends incremental passes in the background and "a single cutover event for all users", because "this approach helps prevent users from updating duplicate copies of content" (team sites migration guide). Chapter 9 freezes per wave, and a read-only source prevents duplicates too. The calendar records which, and why:

COMPARED ON	ONE CUTOVER FOR THE ESTATE	ONE CUTOVER PER WAVE
The freeze	Every wave's final delta and verification at once	One wave's sites
The validators	All of them, inside one freeze	Each wave's, in turn
Between waves	Nothing is split	People work on both platforms, and links between waves need redirects
It fits when	The validators can accept the whole estate inside a freeze its owners accept	They cannot

An attach has no deltas to shrink: its window is the read-only period, and each address cuts over once.

Write the stop before the first wave

Chapter 1 named who can say stop. The calendar writes what they stop on, before the first wave, because after it starts every failure looks like an exception.

A wave's verification fails when any of these holds:

- a `missing` finding nobody can explain, or an `unknown` nobody has read again (Chapter 8);
- on an attach, a failed upgrade session, an error in the upgrade log, or `NeedsUpgradeIncludeChildren` still true (Chapter 23);
- a functional check failed, or a validator who has not signed;
- a rollback, in any state Chapter 9 describes.

One condition is not a failure and stops the next wave all the same: a measured rate that would push it past its freeze or its read-only window.

A stop halts the next wave's freeze and cutover at once and, until the cause is known, every pass that could repeat it. A cause in one site holds that site; a cause in the route, its settings or the farm holds every wave that uses them, and sends the waves already moved back to Chapter 8's comparison. The next wave starts again when the cause is fixed, the fix is proved on the failed wave, and the calendar has moved: later waves shift and never compress, because nobody's hours grew while the project fell behind.

A stop criterion written after the first failure is a negotiation. Written before the first wave, it is a decision somebody already took.

The calendar

One entry per wave, written before the first and signed by the decider Chapter 1 named, with each owner signing the freeze of their sites:

- the sites, as site collections, or as content databases and the web application they serve;
- the group and route from Chapter 29, and the dependencies it waits on;
- each owner and validator, with confirmed hours;
- the window: baseline, passes, freeze, final delta, verification and cutover decision, or backup to verified attach;
- the stop criterion and who may invoke it;
- the rollback boundary Chapter 9 requires.

After each wave, reissue it with what was measured: rates, drift and hours per kind of site. Chapter 35 takes each cutover decision against it.

Out of scope

The pilot and rehearsal are Chapter 31, validation Chapter 34, cutover Chapter 35, rollback Chapter 36 and hypercare Chapter 37; routes, verification and the attach stay in Chapters 29, 8, 9 and 23. Orchestration products, third-party tools, project management, communication plans, pricing and licensing are not covered.

Before you continue

- [ ] Every site in a wave is in use and has a migrate disposition and a route
- [ ] Each wave is sized from rates timed on your own content, not from Microsoft's maxima
- [ ] Copy waves run as parallel tasks per site collection, off-peak, with the tenant planned as one load
- [ ] Attach waves hold every database behind their addresses, the first verified alone and the rest started minutes apart
- [ ] Sites and systems that depend on each other share a wave, or the dependency has a written interim state
- [ ] Every site has a validator whose confirmed hours cover it in its window, hypercare included, and no hours were totalled
- [ ] One cutover or one per wave was chosen with its reason, and every stop criterion was written before the first wave
- [ ] The calendar was signed before the first wave and is reissued after each one with what was measured