Progress®

# Progress® MarkLogic® and Progress® Semaphore™ Life Cycle Policy Guide

Progress® MarkLogic®
Progress® Semaphore™

# Contents

Change Log

| Date | Changes |
| --- | --- |
| November 4, 2023 | Initial version |

# 1 Document Scope

This table explains how every Progress MarkLogic and Progress Semaphore product is covered in this document.

| Products | Refer to |
|---|---|
| Progress MarkLogic Server | Section 2.3 Platform Life Cycle |
| Progress MarkLogic Hub Central | Section 2.4 Tools Life Cycle |
| Progress Semaphore | Section 2.3 Platform Life Cycle |
| All other products | Section 2.4 Tools Life Cycle |

# 2 General Principles and Foundations

## 2.1 Introduction

A product life cycle consists of a series of milestones and activities that determine the different stages of a product release's life cycle: starting from its first General Availability (GA) to the eventual Retirement. The primary objective of this document is to help customers understand the different stages of Progress products release life cycle and to help with planning the upgrade and/or migration strategies of their implementations and integrations.

This Policy is comprised of two main parts.

1. General Principles and Foundations (this section)
2. Product Specific Policy Terms

## 2.2 Platforms and Tools

Progress solutions often include a comprehensive collection of related products and tools designed and validated to work together, and collectively referenced in this document as **Platforms**. The Platform life cycle is described in section **2.3 Platform Life Cycle**. If your product is not a Platform, you can disregard this section.

Alternatively, Progress may ship individual products, tools, connectors, etc. referenced in the document as **Tools**. The Tool life cycle is described in section **2.4 Tools Life Cycle**. If your product is not a Tool, you can disregard this section.

 See Section 1 for information on what is considered a Platform or a Tool.

## 2.3 Platform Life Cycle

**Note:** Please ignore this section if your product is not designated as Platform.

### 2.3.1 Forms of Platform Delivery

#### *2.3.1.1 Platform Releases*

Releases, sometimes referred to as new versions, contain important new product features and enhancements to existing features, along with quality and security corrections.

Each release is available as a complete package to all customers with active maintenance and support services, sometimes referred to as a Service Agreement or a Maintenance contract. Depending on the product, adopting a new release may require modifications to your application (e.g., recompile new and/or modified code to take advantage of new features) or systems (e.g., changes to supported platforms).

Releases are "**Installed**", and customers with prior versions of the product "**Upgrade**" when they move to the newer release.

#### *2.3.1.2 Platform Updates*

Where applicable, updates are used for the resolution of product issues, as determined solely by Progress, including *defects* categorized at Severity 1 (and by exception Severity 2) defined by Progress in Support Service Level Objectives for On Premise and/or *security vulnerabilities* classified at Common Vulnerability Scoring System (CVSS) score level 9 or higher. The delivery timeline and content of an update is always at the discretion of Progress.

Product issues or change requests that don't get resolved in an update will be taken into consideration when planning future releases.

Updates are cumulative and made available to all customers with active maintenance and support services. Updates generally do not require a complete install or migration effort. If that is not the case with a specific update, a separate communication to customers is issued explaining the case in more detail.

Updates cannot be used on their own. Instead, updates are "**Applied**" over already installed releases.

At its discretion, Progress may offer the convenience of rollup updates where the base release and the cumulative updates are bundled.

#### *2.3.1.3 Platform Versioning*

The versioning scheme is used to capture the unique build number. It is most often used when reaching out to Progress Technical Support to verify the exact binaries used by the customer.

### 2.3.2 Types of Platform Releases

#### *2.3.2.1 Innovation Release*

Innovation releases are designed for customers seeking a faster pace of modernization and innovation. Progress Software recommends the use of continuous integration and continuous delivery to help benefit from the faster release cadence. Innovation releases also allow all customers to explore features that will appear in future LTS releases.

Innovation releases enter the Active life cycle phase once they become Generally Available (GA) and move to the Retired life cycle phase immediately upon the subsequent release becoming available.

These releases are ready for production use and are eligible for Updates addressing critical issues. Due to the expected shorter lifespan, the frequency and number (including none) of Updates may be reduced.

#### *2.3.2.2 Long Term Supported (LTS) Release*

From time to time, Progress may elect to designate a new release as Long Term Supported (LTS). An LTS release is meant to be adopted by customers who primarily look for stability and minimal change for their implementation over many years.

LTS releases go through the Active, Sunset, and Retired life cycle phases as described in **Section 2.3.3, Platform Life Cycle Phases**.

#### *2.3.2.3 Technical Preview*

At its discretion, Progress may offer Technical Previews containing **Technical Preview Features** under the terms covered by the applicable End User License Agreement (see Section 3.5.5), and the Progress Customer Validation Program Agreement.

Technical Previews are not recommended for production use.

### 2.3.3 Platform Life Cycle Phases

#### *2.3.3.1 Platform Active Phase*

A platform release enters the Active life cycle phase when it becomes Generally Available (GA) and exits the Active phase when it enters either the Sunset or Retired life cycle phase (see details below).

Active phase releases are fully supported. If problems are found using an Active phase release, Progress will work with customers towards an appropriate resolution if they are covered by active maintenance and support services.

At its discretion, Progress will evaluate Active phase releases for certifications of new third-party products including new operating system versions. Only LTS releases will be considered for ports to new operating systems not previously supported.

**Note:** NO NEW FEATURES will be added, retrofitted, or backported in Updates. New features/enhancements will only be introduced in subsequent Releases.

Progress recommends that customers begin all new projects with the latest Active phase release and migrate or upgrade their existing applications or environments to the Active phase releases as soon as possible. If an LTS release is offered Progress recommends customers always consider adopting the latest Active LTS release version.

#### *2.3.3.2 Platform Sunset Phase (only for LTS releases)*

An LTS release enters the Sunset Phase when both of the following conditions are met:

a) The Target Sunset Date is reached, and
b) A newer LTS release is in the Active life cycle phase.

**Note:** The Sunset phase is not applicable to Innovation releases. Once a subsequent Innovation release becomes Active, the prior Innovation release immediately moves to the Retired life cycle phase.

LTS releases in the Sunset Phase are fully supported and will be evaluated by Progress for certifications on new operating environments, but not for ports to new hardware systems.

If problems are found using product releases in the Sunset Phase, Progress will work with customers towards an appropriate resolution where possible if their licenses are covered by active maintenance and support services. This includes the possibility of an Update at the discretion of Progress.

Progress recommends that customers plan to upgrade from Sunset phase product releases using this document as a guideline.

#### *2.3.3.3 Platform Retired Phase*

Product releases placed in the Retired life cycle phase are not available for sale except for the purchase of additional licenses by existing customers. Retired phase product releases will not be evaluated for certifications on new operating environment versions. Although Updates are not provided for Retired releases, Progress will provide commercially reasonable efforts to resolve

customer issues and answer customer questions on Retired phase product releases covered by active maintenance and support services. However, the knowledge, skills, and development and testing environment required to resolve issues on Retired phase product releases is NOT guaranteed. Progress strongly recommends that customers upgrade to a fully supported product release as soon as possible to maintain the highest level of support for their applications and systems.

**Note:** In the event the product security can no longer be assured, Progress reserves the right to accelerate the previously announced timeline of Release Retirement.

#### *2.3.3.4 Summary of Attributes Available for Each Platform Life Cycle Phase*

| | Active | | Sunset | Retired |
|---|---|---|---|---|
| | LTS Release | Innovation Release | | |
| **Updates** | • | • | • | - |
| **Certify new third-party product version** | • | • | • | - |
| **Port to a new run-time platform** | • | - | - | - |
| **Sales to new[1] Direct Customers** | • | • | - | - |
| **Sales to ISVs for new[1] Customers** | • | • | • | - |
| **License quantity increases** | • | • | • | • |
| **Lifetime Technical Support[2]** | • | • | • | • |

For further information, please contact your Progress sales representative or business partner.

For information about Progress Support Services please refer to

- https://www.progress.com/support

[1] This refers to new product installations for new customers. Sales restrictions of this nature intend to guide new installations towards releases with longer remaining supported life.

[2] As defined in Progress Maintenance and Support Policy. Requires an active maintenance and support services.

## 2.4 Tools Life Cycle

**Note:** Please ignore this section if your product is not designated as Tool.

### 2.4.1 Tool Releases

Progress delivers all new product features, enhancements, modifications, and error corrections in new product *Releases*.

**Note:** Product releases don't receive updates. Instead, old releases are superseded by new releases.

Releases are cumulative and made available as complete installs to all customers with active maintenance and support services.

The delivery timeline and content of every release are always at the discretion of Progress.

### 2.4.2 Backward Compatibility

Backward compatibility refers to the ability of a new product release to seamlessly, or with minimal disruption, replace a prior release installation or deployment.

Backward compatibility is always at the forefront of Progress release planning. While new tool releases are meant to make our products better, some changes inevitably may require additional steps towards a smooth transition.

### 2.4.3 Tool Release Streams

A *Release Stream* is a series of backward compatible product releases. A release stream is denoted by a Release Stream ID as part of the version string.

At any given time, for a given product, Progress may support up to two release streams, an Active Release Stream, and a Sunset Release Stream.

#### *2.4.3.1 Active Release Stream*

The latest releases in *Active Release Streams* are fully supported and will be evaluated by Progress for certifications and if necessary, porting to support new third-party run-time platforms and compatible products.

New product releases introduced within the active release stream are considered in **TOOL ACTIVE** Phase once they reach GA, and all the prior releases in the same release stream are considered in **TOOL RETIRED** Phase.

Active release streams do not have their target retirement date set.

#### *2.4.3.2 Sunset Release Stream*

As products evolve, changes introduced in new releases may impact backward compatibility. When this happens, Progress forms the new active release stream. The prior release stream is then reclassified to *Sunset Release Stream* and its target retirement date is set to one year from the date of this event.

From this point on, the most recent release within this sunset release stream is considered in **TOOL SUNSET** Phase, and all the prior releases in that release stream are considered in **TOOL RETIRED** Phase.

All new releases within a sunset release stream are limited to the resolution of critical product issues, as determined solely by Progress. These issues may include *defects* classified at Severity 1 as defined by Support Service Level Objectives for On-Premise or *security vulnerabilities* classified using Common Vulnerability Scoring System (CVSS) with a score of 9 or higher.

Progress recommends that customers plan to upgrade from the sunset release stream to the active release stream using this life cycle policy as a guideline. Progress will not introduce new sunset release stream releases beyond the announced retirement date.

### 2.4.4 Tool Life Cycle Phases

#### *2.4.4.1 Tool Active Phase*

All new product releases within the active release stream enter the *Active* life cycle phase when they become Generally Available (GA).

Progress will work with customers towards an appropriate resolution for reported or otherwise discovered product issues if the release is in the Active life cycle phase and the license is covered by active maintenance and support services.

#### *2.4.4.2 Tool Sunset Phase*

All new product releases within the sunset release stream enter the *Sunset* life cycle phase immediately when they become Generally Available (GA), i.e., bypassing the active life cycle phase due to lack of backward compatibility.

Progress will work with customers towards an appropriate resolution for reported or otherwise discovered product issues, including new certifications but not porting, if the release is in the Sunset life cycle phase and the license is covered by active maintenance and support services. At its discretion Progress may address issues or introduce product enhancements as part of the new release in the active release stream.

#### *2.4.4.3 Tool Retired Phase*

Both the Active and the Sunset life cycle phases transition to *Retired* phase immediately upon GA of the next release within their respective release stream. The Sunset life cycle phase also transitions to Retired phase when its sunset release stream reaches the announced retirement date.

Customers can continue using releases that have reached Retired life cycle phase for as long as they continue to maintain valid product entitlements (e.g., active subscription or perpetual license). However, with the exception noted below, these releases may be subject to known issues and security vulnerabilities. Progress strongly recommends customers upgrade to Active or Sunset releases available for their respective release streams.

Progress will make commercially reasonable efforts to resolve customer issues and respond to questions about Retired product releases provided the customer has active maintenance and

support services in place. Progress does not guarantee the knowledge, skills, and environments required to troubleshoot and resolve issues on product releases in the Retired phase.

In the event of a technical support case investigation, Progress reserves the right to request the reproduction of the reported issue within the relevant Active or Sunset release. At the discretion of Progress, product changes deemed necessary, including bug fixes, may be introduced in future product releases as part of active or sunset release stream.

**Note:** Retired product releases are not available for sale except for the purchase of additional licenses.

## 2.5 Third-Party Product Support

Progress product releases depend on, interact with, and leverage many third-party products. Their support and compatibilities are covered by the corresponding Product Compatibility Guides (PCG).

### 2.5.1 Supporting New Versions of Third-Party Products

Progress makes reasonable commercial efforts to support new versions of critical third-party products that are announced by their vendors to become GA more than **three** months before a new Progress product release becomes GA.

At a minimum, this commitment covers all supported Operating Systems, Java, and .NET as documented in the PCG.

### 2.5.2 Third-Party Products Retired by their Respective Vendors.

Progress makes reasonable commercial efforts to continue supporting third-party products that, according to their vendors, will not be retired within the first **six** months of a new Progress product release becoming GA.

Plans or intent to discontinue support for critical third-party products are announced in the PCG document in advance. At a minimum, this commitment covers all supported Operating Systems, Java, and .NET.

### 2.5.3 Levels of Third-Party Product Support

At its discretion, Progress may *certify* or *port* its products to work with new versions of third-party products, as they become available.

#### *2.5.3.1 Certification*

Progress performs a series of tests to determine if its products work "as is" with a third-party product version (e.g., Operating System). This does not involve recompilation or otherwise changing the product's source code.

As a result of this process, Progress certifies specific third-party product versions for a given Progress product release. Any limitations or known issues are noted and shared with customers.

#### *2.5.3.2 Porting*

Porting may require Progress to make source code changes to make sure its products work with a specific third-party product version. This is usually done following certification and the resulting findings.

## 2.6 Handling Security Vulnerabilities

Progress takes the security of its products seriously and performs a range of activities to ensure that our products remain secure, including:

- Regular scanning and testing of Progress proprietary code and third-party components.
- Formalized security risk assessment processes
- Active change management practices

Depending on the specifics of a discovered vulnerability and the nature of the risk caused to the affected product release, Progress reserves the right to remedy each security vulnerability in one of the following ways:

1. Provide a new Release or Update with changes that may, if necessary, as determined by Progress, impact a product's:
    a) Functional capabilities (e.g., feature availability or behavior), or,
    b) Non-functional characteristics (e.g., performance or usability)
2. Provide an explanation of the decision to not make changes to the product.
3. Provide instructions to perform configuration-based remediation at either the Product level or the environment level.

In all cases, Progress will provide security vulnerability disclosure in accordance with the published product security guidelines. To avoid dependencies on outdated software that may restrict remediation, Progress encourages customers to keep their deployments current to our latest release and update.

## 2.7 Feature / Functionality Obsolescence Life Cycle

In addition to the product life cycle, features and functionality also move through various phases of maturity from commercial introduction to obsolescence. As features become obsolete, they are handled in one of two ways: they are ***Retired*** or ***Removed***.

- **Retired:** Features/Functionality is identified as obsolete, but not removed from the product.
- **Removed:** Features/Functionality is identified as obsolete and removed from the product.

Security and the backward compatibility of Progress products -based implementations and deployments are two key factors in determining if obsolete features are retired or removed.

Typically, unnecessary, or undesirable features are retired (left in place) before they are removed to give developers time to update their implementations to handle the resulting changes gracefully. In some cases, the decision to remove features is accelerated for security considerations.

Benefits of the Retirement and Removal life cycles phases include:

- Sets the appropriate customer expectations regarding backward/forward compatibility.
- Gives customers enough time to consider and plan changes to their applications.
- Promotes rejuvenation and upkeep of applications, advantageous to partners and customers.
- Better alignment with technology partners such as Operating System vendors.
- Encourages customers to use modern replacement features as appropriate.

### 2.7.1 Retired Features and Functionalities

Retirement provides the ability to identify, communicate and manage obsolescence (and the possible eventual removal) of features and functionality, independent of the products and versions in which they may be included and how they are packaged. Progress recommends that retired features no longer be used. Customers should consider substituting retired features over time with the newer replacement ones. Please note that:

- Retired features continue to function.
- Best-effort support is available for retired features and functionalities.
- Retired features will not include further enhancements.
- Communications will follow the 'Obsolescence Life Cycle Guidelines' as described below.

### 2.7.2 Removed Features and Functionalities

Removal is used where changes in technology or standards have made a feature obsolete or unsafe for use and it is removed from the product. Removed features sometimes have replacement equivalents and typically have no impact on backward compatibility. Key details of removal include:

- Removed features will be removed from the product(s).
- Third party-dependent features, such as run-time platform support, may continue to function.
- There is no support for Removed features.
- Communications will follow the 'Obsolescence Life Cycle Guidelines' as described below.

### 2.7.3 Feature / Functionality Obsolescence Life Cycle Guidelines

The following are the phases for the Retired and Removed life cycle of features:

- Before assigning one of the obsolescence statuses, features that are candidates for retirement or removal will be published to partners and customers for comment, potentially polling for information on the impact that the retirement or removal may cause to current applications.
- Progress Product Management will use the information gathered from this process to assess the obsolescence of each feature or functionality.
- Details about retired and removed features will be included in the Product Compatibility Guide.
- Announcements will be made to inform the Progress community of updates to the status of features.
- Retired and removed features will be identified as such in the Product Documentation.

## 2.8 Product End of Life

Progress is continually re-evaluating its products roadmaps and the existing product line-up to ensure strong alignment with the strategy and the optimal allocation of resources to serve the customers' evolving needs and priorities. Sometimes the difficult decision to end of life (EOL) a product is made, enabling Progress to accelerate the pace of modernization and to introduce new emerging solutions.

The following describes the general impact of a product reaching EOL. For more specific details refer to the corresponding official EOL communication, usually delivered via email.

- Products that are covered by active maintenance and support services at the time of announced EOL will continue receiving Technical Support on the same terms as Retired product releases for the duration of the purchased maintenance and support services.
- Progress does not allow renewal of maintenance and support services for EOL products.
- Once the existing maintenance and support services coverage expires, products with perpetual licenses can continue to be used on an AS-IS basis meaning they will not receive further improvements or fixes from Progress, and no new support cases can be opened for them. The existing product documentation, knowledgebase articles and online user community discussion will remain as the only means of support.

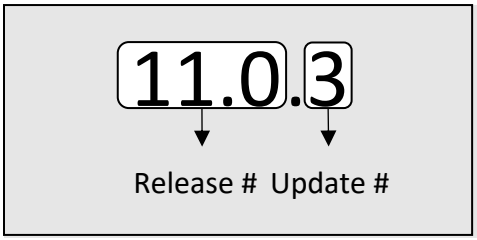
# 3 Product Specific Terms

This section specifies additional terms specific to **MarkLogic** and **Semaphore**.

## 3.1 Versioning

**Note:** The versioning scheme alone may not necessarily identify whether a Release is a Long Term Supported (LTS) Release. As well, individual binary components of the same release may have different build numbers.

Below is an example of the versioning scheme:




## 3.2 LTS Platform Releases

Progress is actively monitoring customer experiences for new Platform GA releases and looks for a representative set of confirmed production deployments. Once this evidence is collected for an LTS release, as established in the sole judgement of Progress, Progress will announce the start of the **four**-year countdown to the release entering the Sunset life cycle phase (*Target Sunset Date*).

The duration of the Sunset Phase for LTS releases is fixed at **two** years, after which the release moves to the Retired life cycle phase.

In the case of Retired Phase LTS releases, there is the possibility of an Update at the discretion of Progress and **for a fee** (based on an assessment of the effort required by Progress), **only for the first year of the Retired Phase**. Starting in year 2 of the Retired Phase, Progress will not consider requests for Updates.

## 3.3 Backward Compatibility

Backward compatibility refers to the ability of a new product release to seamlessly, or at least with minimal disruption, support a prior release deployment. This impacts the operation of applications in use as well as the maintenance and development of these applications.

Backward compatibility has always been at the forefront of Progress release planning. While new Releases and Updates are meant to make the Progress Platform better, some changes inevitably may require additional steps towards a smooth transition. This is particularly important for any changes that may affect mission-critical business applications running in production environments.

**Note:** The complexity of language and the evolution of statistical models in natural language processing (NLP) present significant challenges when it comes to ensuring backwards compatibility at the data level between different versions of the software, even if the application programming interfaces (APIs) are designed to be backwards compatible.

Language is inherently complex: natural language is highly nuanced and constantly evolving. Language is not a fixed entity; it adapts, new words emerge, and meanings change over time. Statistical models (such as deep neural networks), which are at the core of NLP, are trained on vast datasets that capture a snapshot of language at a specific point in time. These models are inherently tied to the linguistic patterns and trends of that time. Any changes to those statistical models, such as fine-tuning or retraining, can lead to shifts in how the software interprets and processes language. When a new version of the software is released with an updated model, it may behave differently when presented with the same input data compared to the previous version.

Users of NLP software may have expectations based on their experience with previous versions. If the software's behavior changes significantly due to updated models or other factors, users may find that their previously effective workflows no longer work as expected.