Crosslight Roadmap

On this page:

Crosslight was engineered from the ground up with the most advanced architecture to achieve holistic goals which includes cross-platform framework, extensible services, data components, great tooling support, as well as comprehensive set of UI components across 4 platforms – iOS, Android, Windows Phone 8 and Windows 8.

This page provides an overview of the Crosslight roadmap and what features you can expect in the upcoming releases. Note that the features listed in this page aren't necessarily targeting the next major release, but spanning across multiple releases throughout the year.
Next major version: 6.0
Estimated release: Q2 2017

New Connectivity and Peripheral Services

The next release will feature new kind of component categories – connectivity and peripheral services. The connectivity services provide cross-platform functionality to connect and send data to a device, as well as monitoring the state of the connection.

Supported connectivity interfaces include:

- Bluetooth connectivity
- Ethernet connectivity

Building upon the connectivity services, there will be a new set of peripheral services and adapters which allow you to connect and work with peripherals in just a few line of code.

Supported peripherals include:

- Printers
- Barcode Scanners
- Card Readers

These services will be easily consumable from shared projects (ViewModel), supporting iOS and Android platforms.

Focused on Performance and Reliability

In addition to many new services, the next Crosslight release will be laser focused on performance improvement, reliability and stability across all components and aspects.

Some key highlights include:

- Core binding engine revamp – the internal data binding mechanism no longer rely on reflections resulting to significant performance improvement.
- Major performance improvements across all platforms – UI, rendering, and code are further optimized to consume less CPU and drawing.
- Memory leaks – all UI components and core services have been tested thoroughly, identified and resolved a number of critical memory leaks.
- Multi-threading support – all data services and core frameworks are now greatly enhanced for robust, thread-safe usage.

As the result, Android and iOS apps built with Crosslight vNext will be approximately 10x faster compared to previous release.

Major Improvements on Crosslight Platform

- Improve view lookup mechanism by implementing resource ID caching which increase processing efficiency and greatly improve performance.
- Enhance *SplitFragment* so that it can have a recycler view fragment as its detail fragment and the recycler view fragment can has it owns toolbar.
- Introduce *Enable Reselect* in Recycler View.
- Add visibility binding to layout container.
- Added device name property to the Crosslight application service context.
- Release any editor focus when user perform click on image picker.
- Improve fragment life cycle to improve performance during navigation.
- Introduce initialize components after view created property.
- Command binding in FormBuilder now works properly after performing rotation or back navigation.

- Support initial selection for FormBuilder's ListView.
- Image loader now support local image file, i.e., content://path.
- Redesigned navigation editor to the latest Material style.
- Added binding support for navigation value in navigation editor.
- Introduce *ShowSeparator* property in FormFragment to display separator for each editor.
- Introduce *ShowSeparator* property in RecyclerViewFragment to display separator for each item.
- Improved *MasterDetailFragment* with new option to enable or disable backstack when navigating from detail container.
- Redesigned ListView editor in FormBuilder, now using the latest Material RecycleView instead of classic list view.
- Support CommandBinding in navigation editor.
- Improved animation timing in dialog presenter.
- Image loader now load resource drawable directly, by passing ImageLoaderTask.
- Stabilize Android activity life cycle manager.
- FormBuilder Picker now support *ListSourceMemberPath*.
- Introduce preferred content percentage for dialog presenter settings.
- Introduce *OnUserInteraction* at AppService class.
- Add *NumericStepper* and *NumericUpDown* in RecyclerView ItemBindingAdapter.
- Support *IsEditing* property binding in recycler view binding adapter.
- Improve item binding and image loader logic in Android – image view that aren't visible in the screen will not be processed.
- Improve *GridView* scrolling performance.
- Introduce *ShouldClose* property in *DialogResult* of dialog presenter.
- RecyclerView selector appearance is now customizable.
- Introduce customizable SelectionView in *RecyclerViewFragment.*
- Introduce *CustomCommandLayoutId* for dialog presenter.
- Enhance *MenuItem* to keep checked state when invalidated.
- Hide keyboard automatically when navigating to search result from *SearchableRecyclerViewFragment.*

Major Improvements on App Framework

- *UserIdentity, User* and *UserService* implementation are now completely replaceable which enable a lot of potential advanced scenarios, such as multi-tenant user management. Several new interfaces are introduced such as *IUser* and *IUserService* which are implemented in these classes by default.
- *UserChangedEvent* delegate event now accepts *IUser* instead of *User* for its payload. There will be minimal breaking changes required in your current application.
- *PushRegistrationService* now supports constructor injection for *UserDeviceToken,* allowing developers to construct custom class to be sent to the server.
- Implement *IPropertyAccessor* for ModelBase for manual property access. You can now implement this interface to write your own property accessor to fine-tune binding performance.
- Enhanced *Synchronization Service* performance: Data that are already existed in EntityContainer will be loaded from the container instead of reloading them from repository.
- Improved *Synchronization Service* implementation, making finalize pending changes process more efficient and robust.
- Improved *Synchronization Service* implementation which automatically perform view query filter against new and updated entities as the result of the data synchronization process. By default, filtered items no longer appear in the view. In previous version, developers are required to handle this case manually.
- Refactor *SynchronizationChannel, SynchronizationServices, EditableLocalEntityRepository, LocalDataRepository,* and *LocalEntityRepository* for extensibility purposes.
- Update EventSource with additional logging info.
- Change *void SetCurrentUser* to *Task SetCurrentUser*, which allows you to process current user with async code.
- Introduce *Delete* method at *IUserSettingsService.*

Major Improvements on Data Services

- SQLite now supports *Include* operation one to one navigation property.
- SQLite now support DateTimeOffset property.
- Introduce *IQueryable.Parse(QueryDescriptor)*.
- Introduce new assembly: *Intersoft.Data.ComponentModel.EntityFramework*, enabling all Crosslight entity services such as *dynamic query* to be consumable in server-side.
- Introduce get property accessor in *EntityBase* class.
- Introduce reset connection state function for SQLite.
- Introduce *Clone* method in EntityBase. This will clone all the data property only.

New Components

- Material Segmented Button for Android, including widget support for Form Builder
- Stepper Control
    - Featuring latest Material UI design

    - Form Builder metadata support
- Currency Editor
    - A specialized number and currency editor designed for business application
    - Realtime formatting as you type
    - Support any cultures and common currency formatting
    - Nullable support
    - Form Builder metadata support

Form Builder

Form Builder has been one of the most powerful features in Crosslight which lets you easily create rich data forms with developer-friendly attribute metadata. In the next release, we see numerous room for improvements that we can do to make Form Builder even more powerful.

In the upcoming release, we are planning to add a host of new, highly-anticipated features that will further simplify form building.

- Dynamic sections – an advanced feature that lets you quickly create highly dynamic forms.
- Platform-specific UI metadata – this new feature will allow you to intuitively customize platform-specific appearance settings which aren't available in the core (shared) form metadata.
- Visibility binding support for Section.
- Introducing a new Dock layout – perfect for adding a sticky header or footer such as image or carousel.
- New built-in editors:
    - Image View – support loading image from local storage as well as remote source.
    - Segmented Button – a convenient selection control.
    - Grid View – a gorgeous list view control that presents items in intuitive manner, perfect for e-commerce and B2C apps.
- New extension editors:
    - Map View
    - Carousel View
    - Calendar View

WatchOS 3 Support

The upcoming Crosslight release will include full support for WatchOS development. Beyond just native API, Crosslight 5 will live up to its promise, enabling you to create Watch apps with your 100% existing UI logic. This means that your current Crosslight shared project that target iOS, Android and Windows – can be entirely reused to target Apple Watch without code changes.

Key features include:

- MVVM-enabled Interface Controllers
- MVVM and binding support for all native Watch UI controls including label, button, image, etc.
- Command and two-way binding support for buttons and input controls
- Advanced table binding support (which aren't available in native WatchKit itself):
    - Works in the same way and manner with existing Crosslight binding. Just drop in a table view in Watch project and given an ItemsSource in the binding, it will automatically display exactly the same items as in other supported platforms.
    - Support smoothly animated collection changes.
    - Support grouping and collection group changes.
    - Support detail navigation.
    - Support single and multiple selection.
    - Support image view.
    - Support advanced image view with customizable size and circle radius.
- Supported Crosslight Services
    - All non-UI services and frameworks are all supported, including view service, data access framework, sync framework, cryptographic, image loader, etc.
    - Certain mobile/UI services will be supported per availability in native WatchOS 2.
- Navigation Services
    - Support both Push and Modal navigation
    - Support Watch's unique page-based (carousel) navigation
    - No changes in Crosslight navigation API, means your existing navigation logic will continue to work as is.
- Presenter Services
    - Support all Presenter lineups:
        - Message Presenter
        - Toast Presenter
        - Action Presenter
        - Activity Indicator Presenter
- MVVM and binding support for specialized UI Components
    - Maps
    - Picker (Digital Crown)
- Advanced features:
    - Efficient memory and resources consumption, with ability to detect when an interface controller is no longer active in the

screen.
- Conditional logic targeting Watch devices and its kind.
- Support contextual menu binding to command (Force Touch).
- Support MVVM and data binding for Glance interface controllers.

If you have feature suggestions not listed in this roadmap, please feel free to post your suggestions to Crosslight Community. We will be happy to review your suggestions and include it in our roadmap whenever possible.

Delivered in Crosslight 5 Release

On February 2016, Crosslight 5 was released with more than 20.000 new API, deliver nearly 80% of the features defined in the 2016 roadmap. Exceeding expectation, the release also includes many exciting new features not defined in the roadmap which we believe are important to be included in the milestone. The following list shows the completed features in Crosslight 5.

- Brand-new Crosslight Android Material Library with 20+ material-design user interface components
- Streamlined Crosslight development with enhanced NuGet packages support
- Introduced 50+ Crosslight NuGet packages with multiple platform targeting
- Two new charts – gauge and circular chart
- New signature pad component
- Major improvements to Calendar component
- iOS 9 support
- 100+ core API for even easier and simplified MVVM development

To see the complete new features introduced in Crosslight 4, see Crosslight 5.0 Release Notes.

Related Topics

- Crosslight 5.0 Release Notes
- Crosslight 4.0 Release Notes
- Crosslight 3.0 Release Notes
- Crosslight 2.0 Release Notes
- Crosslight 1.0 Service Pack Release Notes