

New in 7.3

Merge of Servoy 6.1.x into Servoy 7.3

With the release of Servoy 7.3 the upgrade path for the 6.1.x branch will be the Servoy 7.x branch, as Servoy 7.x can be considered Servoy 6.1.x + Servoy Mobile. This means that Servoy 6.1.6 is the last update in the Servoy 6.1 branch and that installations that run Servoy 6.1.x can upgrade to Servoy 7.3, skipping 7.0.0 - 7.2.

The auto-update mechanism in Servoy Developer and the command-line upgrade mechanism in the Servoy Application Server will start offering the Servoy 7.3 version as update on Servoy 6.1.x releases as of November 8 2013.

For more information on all supported version, see [Supported Versions](#)

See the following links for the Release Notes of previous 7.x.x releases:

- [New in 7.0](#)
- [New in 7.0.1](#)
- [New in 7.1](#)
- [New in 7.2](#)

Mobile Client

- [SVY-4231](#) Added online mode to the Mobile Client, allowing to display data retrieved live from the server
See [Online Mobile](#) for more details
- [SVY-4908](#) Added support for debugging Mobile Solutions
See [Debugging Mobile](#) for more details. As a result of gaining this new debugging ability, the emulated Mobile Debug Client has been removed from Servoy Developer
- [SVY-4258](#) Added option to specify the name of the Service Solution when exporting the Mobile Solution
- [SVY-4258](#) Added `application.get/setServerURL()` to specify the URL of the Application Server to which the Mobile Client will connect to access its Service Solution.
If the server URL is not set at export, it must be set before `security.authenticate` is called from custom login form. This way, the URL is set before any calls to server are done
- [SVY-4402](#) Added option to specify a custom login form
On mobile solution properties, a login form can be specified. This login form has to call `security.authenticate` to authenticate server side against a username/password. After `security.authenticate` is called the flow of application will be the same as for default login screen.
- [SVY-4799](#) Added `application.getApplicationType()`
- [SVY-4884](#) Added support for using `FoundSets` based on in-mem datasources for offline data
- [SVY-5011](#) Added option to programmatically logout
- [SVY-4672](#) Added `JSFoundSet.getRecordIndex(JSRecord)`
- [SVY-5088](#) Added `plugins.mobile.getUUIDPKValueAsString`
When communicating between the Mobile Client and the Service solution, records need to have a UUID PK. Inside the Mobile Client however, the UUID's are translated into unique integer values, to preserve memory and storage. The `plugins.mobile.getUUIDPKValueAsString` method allows to get the UUID PK value for a specific `JSRecord`. This is needed in case callbacks are made from the Mobile Client to the Service Solution outside the the built-in sync mechanism and information about specific records needs to be communicated.
- [SVY-4397](#) Refactored text property on Bean to `innerHTML` and also made it available `@designTime`
- [SVY-4657](#) Added `new/removeValueList()` to `solutionModel`
- [SVY-5036](#) Added `application.get/setUserProperties` support to the Service Solution
When setting User Properties in the service solution using `application.getUserProperty()`, the properties and their values will be send back to the Mobile Client where they are persisted. With each subsequent request to the service solution the properties and their values will be send back to the service solution, where they can be accessed through `application.getUserProperty`. Note that the sending of the properties and their values back to the service solution with each request is only applicable to requests made by the Servoy API. In custom AJAX calls done through JQuery for example, the special HTTP Request Headers that take care of sending the properties and their values to the service solution are not set and thus the properties and their values are not available in the service solution.

Servoy Developer

- [SVY-4567](#) Added support for the standard GIT setup in Servoy Developer
Servoy Developer now also supports working with projects (== solutions) that don't contain all the project files inside the project directory in the workspace, but in a different location, like a GIT Repository which is located somewhere else on the file system
- [SVY-1321](#), [SVY-4744](#) Script editor now supports `@typedef` (<https://developers.google.com/closure/compiler/docs/js-for-compiler>) where one can have a variable that is then an alias for a more complex type. The variable name can then be used in `@param {varname}` or `@type {varname}`
- [SVY-4548](#) Support for `@protected` annotation on members of constructor functions
- [SVY-2826](#) Added a Code Template for `FoundSet` loops
- [SVY-4893](#) Made `application.output()` statements with WARNING and ERROR level appear in red in the Console in Servoy Developer, so they stand out more
- [SVY-4964](#) Added `i18n` key selector to dataprovider title editor

Solution Development

**Behavior Changes**

Due to the implementation of [SVY-5036](#): 'Added application.get/setUserProperties support to the Service Solution', the behavior of get/setUserProperty in the Headless & Batch Processor clients has changed.

- Before Servoy 7.3 User Properties set from within Headless Clients would be persisted in the servoy.properties file of the Application Server and be shared between all Headless Clients running on that server.
As of Servoy 7.3 the behavior changed in the following ways:
 - In most Headless Clients the User Properties are stored on Session level, which means that when the session ends, the User Properties are no longer persisted. This also means that User Property values are not shared anymore between the different Headless Clients
 - In Headless Clients used by the Restful plugin the User Properties are added as an HTTP Header (name=servoy.userproperties) of the generated HTTP Response for each call to the Restful API. The client that calls the Restful API (a browser for example) can read the values from the HTTP Header in the response. In order to persist the User Properties between subsequent calls to the Restful API, the client that calls the Restful API needs to add the User Properties as to the HTTP Request header.
- The only exception to this rule are Headless Clients created through the Headless Client Java API with an HTTP Request/Session, for example when integrating the Headless Client into JSP pages. In this case the User Properties were stored in the HTTP Session object and still are

- [SVY-937](#) Encapsulation support on ValueLists, Relations and Media
- [SVY-3259](#) Made the Form Encapsulation property clearer
- [SVY-4734](#) Support to deprecate Forms, ValueLists, Relations and Media's
- [SVY-1873](#) Added option to launch a Smart Client without initial MenuBar/ToolBar/StatusBar
- [SVY-4869](#) Ability to specify the PK column in databaseManager.createDataSource()
- [SVY-4547](#) Added support to the RESTfull WebServices plugin to receive multi-part post
- [SVY-4281](#) Added option to retrieve fallbackValueList on JSValueList
- [SVY-2003](#) Added option to open files in the Web Client, instead of downloading them
- [SVY-4799](#) Added APPLICATION_TYPES.MOBILE
- [SVY-1088](#) Added support to call databaseManager.saveData(JSRecord/JSFoundSet) from within table events on related records
This allows the creation and successive save of related records when a parent records gets inserted/updated
- [SVY-3611](#) Exposed formIndex property on elements
- [SVY-4504](#) Added JSDataSet.getColumnNames()
- [SVY-2241](#) Added JSComponent.getDesignTimePropertyNames()
- [SVY-1978](#) Support for working with literal values in relation through the solutionModel

Web Client

- Several updates to the rendering of the TableView to improve performance

Deployment

- [SVY-1718](#) Tomcat setting "useHttpOnly" now set to true by default in order to mitigate possible XSS attacks
- [SVY-4677](#) Added option to remove Profiles through the Servoy Admin pages
- [SVY-3851](#) Option to disable Smart Client usage when deploying as a WAR

Public Java API

- [SVY-3192](#) Added several methods to the IPageContributor interface to add JavaScript and CSS resources globally to all current and future windows and dialogs in the current Web Client session

Open Source

- Updated the Open Source guide with instructions for Servoy Mobile