New in this release

In This Chapter

Highlights

The Servoy 6.1 release focuses on the following area's:

- Extending the capabilities to create compelling User Interfaces
- Web Client compatibility
- Developing large-scale Solutions
- Ease of deployment

User Interface enhancements

Servoy 6.1 introduces several new UI widgets and extended styling capabilities

UI Widgets

- Listbox: A ListBox is similar to a Combobox, but instead of having a dropdown to display all the possible values, it has a multi-line UI where displaying the possible values. The ListBox is availble in two variations: Single and MultiSelect
- Accordion Pane: The AccordionPane is similar to a TabPanel, in that it contains multiple Forms, of which only one can be shown at the time.
- Spinner field
- Popup Form

Styling options

- CSS3 border-radius
- CSS3 gradients
- CSS3 RGBA colors (semi-transparent colors)
- CSS background-image
- TableView & Portal header styling through CSS
- TableView & Portal odd/even/selected row border styling through CSS
- Form Part StyleClass support
- Undecorated JSWindows
- Support for Smart Client Look & Feels to decorate the window chrome

Web Client compatibility

- ListView: The ListView mode for forms is now fully supported in the Web Client
- Title Border: The TitleBorder is fully supported on Forms, Panels, Labels and Border Panel shape
- Shapes: The Shape widgets (Horizontal/Vertical Line, Circle, Rounded Rectangle and Border Panel) are now fully supported in the Web Client

Programming API

- Multiple TopLevel scopes
- Entity scopes
- Formatting on Labels
- QueryBuilder API
- Folder support in the Media library

Deployment

- Support for installing the Servoy Application Server on any compliant Java Application Server, like JBoss, WebSphere, GlassFish etc. using WAR files: Servoy Developer 6.1 features the ability to export an entire Servoy Application Server plus a selections of plugins, beans, lafs and drivers as a so-called WAR file, which can be deployed to any compliant Java Application Server
- MetaData Export & sync facilities
- JMX Application Server monitoring
- Extended SQL Performance data on Servoy Admin page
- Support for updating to a specific buildnumber

- Browser selection when launching a Debug Web Client

Servoy Developer

Solution Explorer

- Changed the display of overloaded functions: Instead of showing an overloaded function once, it now is listed in all it's variations
- Aligned the way Constants are exposed under nodes on the Solution Explorer
- Icon for Constant nodes in the Solution Explorer
- Added "Add Module" & "Create new Module" options to the context menu of the Active solution and it's existing modules
- Rename Form dialog opens with the current name pre-filled
- Added "Open SQL Editor" and "Search for References" menu options to the context menu on Views
- Better warning when deleting a Table with the option to cancel the delete
- Debug menu item on methods is now always enabled and will show the user a warning when trying to debug without having a debug client running
- Added "Search for References" for dataproviders: Available through the context menu on selectedRecord and relation nodes in the Solution Explorer and on the Table Editor
- Module names are added behind relations shown under a form node, to align the behavior with other locations where relations are shown in the Solution Explorer
- Added the ability to move calculations and aggregates to different module
- Renamed the "delete database" option to "Delete database connection"
- Warning that prevents Export of Active Solution due to errors in the solution is now shown immediatly when starting the export wizard, instead of at the end
- Prevent creation of tables prefixed with "servoy_"
- Improved the Synchronize DB Tables with DB information wizard:
    - the actions Ignore/Create/Delete can now be applied to a selection containing multiple rows, instead of just one row
    - Relabeled the "Ignore" option to "Skip"

Build System

- Added builder markers on non-externalized text
- Added option to suppress individual builder markers for non-externalized text using //$NON-NLS-$ comments at the end of the line of code. See the Quickfix option on builder markers for non-externalized text
- Added builder markers for use of deprecated designtime properties
- Added builder markers for unused private variables
- Added builder marker for non-existing parent form (extendsForm property on Forms)
- Added builder markers for element name conflict when working with Form inheritance
- Added builder markers for invalid Foreign Type entries on columns
- Added builder markers on methods and variables having both the @private and @protected JSDoc tags
- Added builder markers for duplicate scope names
- Added preferences for all non-JavaScript builder markers, see Window > Preferences > Servoy > Errors/Warnings
- Many enhancements to the mechanism to perform an automatically rebuild when changes are made to the design of a solution that can affect the generated builder markers

JSDoc

- Added 'nls' option to the JSDoc @SuppressWarning tag to suppress all builder markers related to non-externalized text within one method
- Added 'unused' option to the JSDoc @SuppressWarning tag to suppress builder markers related to unused private variables

Editors

- Table Editor: Added sorting on the events tab
- Table Editor: Made rows representing excluded columns visually stand out by using a light gray font color
- Script Editor: Automatically add type info on @param JSDoc tags on onDataChange event handlers based based on teh dataprovider type when creating the onDataChange handler on an element tied to a dataprovider
- Script Editor: strike-through formatting of deprecated API, see Window > Preferences > JavaScript > Editor > Syntax Coloring > Core > Deprecated Members
- Script Editor: QuickFix support (Control-1 or context menu on builder markers) for builder markers related to non-externalized text in .js files
- Script Editor: Quickfix support for usage of undeclared variables
- Script Editor: Added Save action to cleanup unused $NON-NLS-$ tags on save, see Solution Explorer > Solution/module > context menu > Properties > JavaScript > Editor > Save Actions > Externalize Strings
- Script Editor: Bookmarks & Task support in the context menu of the ruler of the Script Editor
- Form Editor: Improved the Parts tab to also show inherited Form parts
- Form Editor: Added "Place as Labels" options to Portal Placement wizard
- Form Editor: Added option to "Save as Template" dialog to optionally group the elements or not
- Form Editor: Added wizard to place Accordion Panes

Misc.

- Added option to select the browser in which the Debug Web Client gets opened. The list of available browsers is managed through Eclipse's browser support: Window > Preferences > General > Web Browser
- Added global preference to define the Sequence Type used for new Tables, see Window > Preferences > Servoy > Table Creation Settings
- Added global preference to control the value of the navigator property on new forms.
- MetaData synchronization & export. See MetaData export for additional information
- Made the Launch dialog for Debug Clients non-modal

- Removed warning dialog when editing Calculations in Developer while a Debug Client is running
- On Linux default to Metal LAF to prevent stalls of Servoy Developer
- Added option to include the DataProvider or element name in the name of newly created event handlers for form elements and table events, see Window > Preferences > Servoy > Automatic Naming Strategies
- Aligned label text used in the IDE for referring to the "Externalize Strings" functionality
- Made "Call Hierargy" view open by default below the Editor area
- Added HTML Preview to text editor dialogs
- Renamed "Form Editor Options" under Window > Preferences > Servoy to "Servoy Design Perspective options"
- Made the results of a Solution Import the last page of the wizard, instead of a separate dialog

Solution Development

User Interface Widgets

- Added a Single and MultiSelect Listbox element. A ListBox is similar to a Combobox, but instead of having a dropdown to display all the possible values, it has a multi-line UI where displaying the possible values. ListBoxes can be targeted for styling through CSS using the "listbox" selector
- Added a Accordion Pane element.
- Added a Spinner displayType for fields
- Added Popup Form: a Popup Form is similar to showing a form in a dialog, but with a few unique behaviors:
    - The Form is shown without any dialog chrome
    - When the user clicks outside the Form that popped up, the popup will automatically be hidden
    - The scripting API through which the Popup Form is shown and hidden allows the developer to specify which dataprovider needs to be updated
    The PopupForm feature is part of the window plugin. It's API consist of:
        - plugins.window.showFormPopup()
        - plugins.window.closeFormPopup()
        - plugins.window.cancelFormPopup()
        See the sample code for more information

- Added formatting support on labels
- Enhanced rendering of rounded Rectangle and Circle shapes in the Smart client

Styling options

Servoy 6.1 offers the ability to take advantage of some of the new styling features that are made possible through CSS3.

In the Web Client, Servoy relies on browser support for the relevant CSS properties for the implementation of these new styling capabilities. This means that some styling features might not be supported on older browsers. Where logical, Servoy supports the specification of fallback values. For example when using semi-transparent background colors, it's also possible to specify a solid color as a fallback value for browsers that do not support semi-transparent color values. Where browser vendors use different CSS syntax to achieve the same end result, Servoy takes care of generating the required CSS syntax for the different browsers.

In the Smart Client large part of the styling of elements in controlled by the Look and Feel with which the Smart Client is running. Support for the new styling features in the Smart Client is largely limited to styling Forms and Form Parts.

Feature	Description	StyleSeet support	Designtime support	Web Client support	Smart Client support
Rounded Borders	Ability to have borders with rounded corners.	Yes, through the CSS3 border-radius property Property Syntax: - border-radius: radi: px|% - border-radius: horizontal-radi:px|% / vertical-radi:px|% Examples: - border-radius: 5px - border-radius: 5px 10px 5px 10px / 10px 5px 10px 5px; - border-radius: 5px; - border-radius: 5px 10px / 10px;	Yes, using the Rounded Border borderType	Yes, depending on the browser support for the border-radius property	Previous version of Servoy already supported Rounded Borders through the SpecialMatte borderType value. Additionally there is now also a RoundedBorder borderType value, which offers more flexibility. Due to limitations in the Look and Feels used within the Smart Client, background-colors of elements bleed out of a rounded border and/or LAFs have a border hardcoded into the Look and thus do not render the Rounded Boder or render both.
Gradient backgrounds	Ability to have background color transition from one color to another color. Support is currently restricted to linear gradients with a start and end color and a vertical or horizontal orientation	Yes, through the linear-gradient value for the CSS background-image property Property Syntax: - background-image: linear-gradient(,,); Examples: - background-image: linear-gradient(top, white, gray); - background-image: linear-gradient (left, #fefcea, #f1da36);	No	Yes, depending on browser support. With the current restrictions in place, the browser support is good, including older IE versions	Only on Forms and Form Parts and only horizontal gradients

Semi transparent colors	Ability to use colors that are not 100% opaque, but semi transparent, which means that whatever lies underneath partially shines through	Yes, using rgba values instead of hex of rgb values.
		
		Value Syntax:
		
		• rgba(red:0-255, green:0-255, blue:0-255, opacity:0-1) Examples:
		• color: rgba(255, 0, 0, 0.5) //50% transparent red
		• background-color: rgba(0, 255, 0, 0.1) //10% transparent green
		• border-color: rgba (0, 0, 255, 0.9) //90% transparent blue
			No	Yes, depending on browser support	No

Opacity	Ability to make an component and all it's children render semi-transparent	Yes Property Syntax: • opacity: 0.0-1.0 Examples: • opacity: 0.5 //the component and it's children all become 50% transparent	No	Yes, depending on browser support	No
Background Images	Ability to have the background of a Form, Form Part or element rendered using an image. The image can optionally be repeated horizontally, vertically or both The image can be set in a certain position The image can be fixed in position within a scrollable panel The image can be resized before being used	Yes Property Syntax: • background-image: url (media:///myImage. png); • background-position: vertical: top|center|bottom|px |% horizontal: top|center|bottom|px |%; • background-repeat: repeat|repeat-x|repeat-y|no-repeat; • background-attachment: fixed|scroll; • background-size: px|% px|%;	No	Browser support for background-image, background-position, background-repeat and background-attachment is good, as they are part of CSS 2.1. Support for background-size is restricted to modern browsers, as it's part of CSS3	Only on Forms and Form Parts

- TableView & Portal header styling through CSS: Headers of TableViews and Portals can now be targeted from StyleSheets using the grid_header selector. Supported CSS properties are: border-*, color, background-color and font-*
- TableView & Portal odd/even/selected row border styling through CSS: Added support for the border-* properties in the odd/even/selected selectors in StyleSheets used for styling rows
- Form Part StyleClass support: Form parts can now be targeted from StyleSheets using the selectors title_header, header, body, footer and title_footer. Form Parts now also have a designtime StyleClass property. Supported CSS properties: background-*
- Undecorated JSWindows: The JSWindow object supports an "undecorated" property. When set to true, the Dialog or Window will not have the default chrome. Note: In the Web Client the undecorated property is not supported for JSWindows of type JSWindow.WINDOW, as it's not possible from a webpage to control how new windows are displayed. In most browsers a new window will be an additional tab.
- Support for Smart Client Look & Feels to decorate the window chrome
  Java Swing Look and Feels that can be used to provide a global styling to Smart Clients can now be allowed to also decorate the chrome of windows and dialogs. Support for this needs to be explicitly enabled through the "servoy.smartclient.allowLAFWindowDecoration" setting on the Servoy Admin page. support is disabled by default, because on some platforms it can cause a flickering UI while resizing the window or dialog. This is due to a longstanding Java Swing bug that isn't likely to get fixed anymore: http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=5079688

Web Client compatibility

- ListView: The ListView mode for forms is now fully supported in the Web Client
- Title Border: The TitleBorder is fully supported on Forms, Panels, Labels and Border Panel shape
- Shapes: The Shape widgets (Horizontal/Vertical Line, Circle, Rounded Rectangle and Border Panel) are now fully supported in the Web Client
- Added Label rotation support. Relies on browser support for the CSS3 text-rotation property

Programming API

- Multiple TopLevel scopes: added the ability to define additional top level scopes besides the standard "globals" scope. Scopes are accessible in scripting through the "scopes" keyword. Variables defines in scopes can be used in relations
- Entity scope: added a scripting scope on datasources (tables/views in a database server) to create methods. The methods will be available on any foundset object that is based on the datasource or can be used as event handlers for table events
- Added additional Table Events:
  - onFind & afterFind
    Methods attached to these event get called when foundsets based on this table attempts to go into findmode. The onFind event handler can prevent the foundset from entering findmode by returning false. The afterFind will not be called in that case
  - onSearch & afterSearch
    Methods attached to these event get called when foundsets based on this table attempts to exit findmode. The onSearch event handler can prevent the foundset from leaving findmode by returning false. The afterSearch will not be called in this case

- onRecordCreate & afterRecordCreate
  Methods attached to these event get called when an attempt is made to create a new record in a foundset based on this table. The onSearch event handler can prevent the foundset from leaving findmode by returning false. The afterSearch will not be called in this case
- Formatting on Labels: Support added to specify formatting on labels. This is particularly useful when using labels to display data in grids on order to utilize the onDoubleClick of labels to perform an action when the user doubleclicks a record.
- QueryBuilder API: allows for the creation of SQL statements in an object oriented way, without String concatenation. The generated SQL Statement objects can be used in loadXxxxRecord functions and databaseManager.getDataSetByQuery/getFoundset
- Folder support in the Media library: the contents of the Media library can now be structured in directories
- Added option to specify explicit DataProviderID's for columns of tables: when specified, the column will be available in scripting and form design under the specified name, instead of the actual column name
- Added support for literals in relations

Misc.

- Added option to specify arguments for the callback on plugins.window.createShortcut
- The new .storeBounds property on JSWindow instances (present since 6.0.5 with default value "true") now defaults to "false". It specifies whether the bounds of the JSWindow are stored or not when the window is hidden and then used again when the window is shown again (in the same or subsequent sessions).
- Deprecated all runtime "allxxxx" properties. Use the solutionModel instead
- Added support for injecting LINK tags through non-editable HTMLArea's in the Web Client, with a href attribute containing a media library url
- PDF Forms Plugin: Added option to load templates from any location (backported to Servoy 6.0.5)
- Added support to create relations through the solutionModel with names that contain both upper/lowercase letters. The name converted to just lowercase characters needs to be unique within the solution. in scripting the name needs to be references with the correct casing.

Behavior Changes

- Subtypes of plugins (for example JSFile of the file plugin) must now be prefixed with their plugin name when used in JSDoc for typing, to prevent clashes between different plugins having sub types using the same name. Example: use @type {plugins.file.JSFile} instead of @type {JSFile}
- The type JSFoundset was deprecated and replaced by JSFoundSet for consistency
- The behavior of the onRender event is changed, so that it is no longer needed to explicitly handle both the default and exception scenario. Only handling the exception will be sufficient from now on. For example when setting the fgcolor on fields displaying negative numbers to red, prior to this release the onRender logic would need to set the fgcolor for both negative and positive numbers. As of this release only the exception, being the negative numbers needs to be handled. So only when the number is negative does the fgcolor need to be set. If the value is not explicitly set in the onRender, it will render with the default value automatically.

- Rounding of numbers in utils.numberFormat changed to use round_half_up mechanism
- DBTree(Table)view bean: row highlighting highlights the entire row
- Form.onRecordSelection is also fired when the selection is being set to none (-1) if an empty foundset is set. (so foundset.getSelectedRecord() can return null)

Deployment

Behavior Changes

- As of Servoy 6.1 access to tables in the Servoy Repository is restricted by default for security reasons. The "servoy_client_stats" table that resided in the Servoy Repository is now deprecated. Instead of the "servoy_client_stats" table the session tracking is now done in the "client_stats_ table in the database server marked as Log Server. If there is no Log server defined or it does not contain a table named "client_stats", either the old "servoy_client_stats" in the Servoy Repository will still be used if it exists or otherwise there will be no session tracking information recorded.

WAR export & deployment on J2EE standard compliant Java application servers

Servoy 6.1 offers an alternative approach to running the Servoy Application Server. Prior to Servoy 6.1 a Servoy Application Server could only be installed using the All-in-One installer, which would install the Servoy Application Server, bundled with Apache Tomcat.

As of Servoy 6.1 it is possible to create a so-called WAR file from Servoy Developer, which contains the Servoy Application Server logic, including selected plugins, beans, drivers and LAFs, but excluding the Apache Tomcat application server and PostgreSQL database. The exported WAR file can then be deployed on any J2EE standards compliant Java application server, like JBoss, Glassfish, Apache Tomcat, Jetty or IBM WebSphere for example.

For detailed information, see WAR Export & Deployment

MetaData export

Servoy 6.1 offers facilities to mark certain tables as containing Metadata for the solution. When marked as such, the data from the table can be made part of the design of the solution and thus can be synced to a Team Provider like SVN.

When exporting a solution to a .servoy file, the export wizard in Servoy Developer has an additional setting to include Metadata. On import of the Solution on an Application Server, the Metadata exported with the Solution gets merged with existing data in the database(s) connected to the Application Server. For more details see Metada data import

Servoy Application Server update mechanism

In addition to the already existing options for the command-line servoy_updater.jar, a new option is introduced to allow upgrading to a specific buildnumber. The difference with the already existing options is that it allows updating to a specific release of Servoy, not being the latest release. The new option is available through the -releaseNumber {number} parameter.

This brings the the total of command-line parameters to three:

- -beta : updates to latest beta release
- -intermediate number : updates to release number, if that release exists
- -releaseNumber number : updates to release number, if that release exists and is not beta

Servoy Admin page

- Added option to set the idle timeout on database servers. This setting controls the interval after which unused/idle connections are cleaned up
- Added option to manage security groups
- Added ClientID & solutionName to Server Log
- Added more details to the Performance Data log. Click the "Avg. Time" column header to see the additional details
- Renamed "Query validation Type" to "Connection Validation Type"
- Fixed the position of the left-hand side menu
- Moved "Flush All" and "Delete All Solution" options on the Solutions page into the header of the grid

Misc.

- Caching improvements for unstored calculations depending on other calculations
- Web Client performance improvements by reducing the generated markup for TableViews and Portals
- JMX support for monitoring the database server connection pools of the Servoy Application Server

Misc.

- Updated to Rhino JavaScript engine 1.7R3
- Updated to Eclipse Indigo (3.7.2)
- Updated to Eclipse EMF 2.7.1
- Updated to Eclipse GEF/Draw2D 3.7.1
- Updated to YUI 2.9.0
- Updated to PostgreSQL 9.1.3
- Updated to Java Mail 1.4.5

Public API

Servoy 6.1 introduces a improved mechanism to provide documentation on plugins. While the old mechanism will still work, plugin developers are advised to adopt the new mechanism. See Replaced the IScriptObject with IScriptable, making it possible to document the API of plugins through JavaDocs for more details on this change.