There and Back Again

A Children's Bedtime Story by Marvin Magix

New York Rhymes Bestseller;
"The Doorway to the Rabbit Hold for the 3rd Millennium Man"

PS!
Don't Panic!

Index

Find the name of the chapter you'd wish to read up on here, before diving into the 'big pie' ...

1. Welcome to Magix!

Truly a Strange and Wonderful World ...

A place where you can become literate in regards to computers. A place where you can express yourself, creating what you want out of your computer. A place where you are in charge, a place which is fun!

This is your First tip of the day. Leave These Tips On to make sure you get Useful Tips and Tricks as you proceed deeper and deeper into the Rabbit Hole ...

In fact, your first Tip is to use the Arrow Keys in the Top/Right corner of this window to navigate forward and fast read the next upcoming 5-10 tips. They are all crucial for getting started with Magix ...

If you need to re-read a previous tip, you can click the previous button ...

4 sec Intro

Magix consists of two majorly important pre-installed modules; The 'Publishing' System and the 'MetaType' System.

Publishing is where you go to create your WebSite, and MetaTypes is where you go to create Applications. Though, really the difference is more blurry than you think. For instance; It is difficult to show any Applications to your end users, if you do not have any Pages to 'host' your Application ...

So Pages can be thought of as Views in your Applications if you want to, or your entire hierarchy of Pages can be viewed as a gigantic plugin Application, which it actually in fact is ... ;)

It is actually quite useful to stop separating between 'Old-Time Constructs' such as 'code', 'data', 'input' and 'output'.

Old World thinking, trying to categorize things into different types, are much less useful in Magix than what you think. In Magix, everything is kind of 'mushy'. Or 'everything is everything' I guess you can say. This is what makes it possible for you to Stay in Control and deliver Secure and Stable Systems, regardless of the Complexity of your Domain Problem ...

We recommend people to Start with Learning Publishing and how the WebPages work. Then later, only when a firm grasp of Pages and Templates are understood, we recommend moving onto Applications ...

Basics ...

But before we can do anything else, we need to learn the Basics ...

Most of Magix is made up of 'Basic Components', which are tied together to create a whole.

For instance, a button will mostly look the same everywhere. By default a button will be Gray and use Bold, Black and Big Fonts ...

A good example of a Button is the Top/Right corner of this tooltip, which has two Buttons. One Paging forward, and another paging Backwards in the Hierarchy of Tips and Tricks ...

A 'Grid' is when you see a list of items. A good example of a Grid would be MetaTypes/Meta Actions ...

There are many types of 'Basic Components' like these in Magix. Over the next couple of pages, we'll be walking through some of them which you'll need to understand to be able to get the most out of Magix ...

Grids

Most Grids in Magix have tons of features. These features includes; Paging, In Place Editing of Values, Filtering, and so on ...

To Filter according to a Column, all you've got to do is to click the header of your Grid, choose which type of Filter you want to apply, type in its value, and click OK ...

The arrow buttons, normally at the bottom of your grids makes it possible to traverse forward and backwards in your list of items. The double arrows takes you 'all the way' in its direction ...

Open up Meta Types/Meta Actions and play around with that grid by filtering, creating a couple of new items and so on.

Make sure you don't change any of the existing items, since some things are dependent upon 'System Actions' which must be defined for your system to properly work ...!

To see Paging you'll normally need to have more than 10 items in your grid. To see 'all the way paging', you'll normally need more than 20 items ...

Make sure you also click the 'Edit' column. Sometimes this column will say 'Edit' while sometimes it'll show a number like in the Action view. However, clicking the Edit Column, will always somehow bring you to a View where that object can be edited in 'full version' ...

Some things should be very similar for mostly all grids like this in Magix. For instance ...

Clicking the '+' button will almost always create a new object of that type ...

If the Text of a Grid Cell is Blue, this means that you can edit the value directly by clicking the Blue Text, which will exchange it with a 'textbox', from where you can edit its value ...

Publishing ...

A Website consists of Pages. Every Page is the equivalent of one 'URL'. Although URLs doesn't really exist in Magix, it helps to think of a page as such. Beside, creating a URL based Navigation Plugin would be piece of cake anyway, due to the Architectural Principles Magix is built on ...

Anyway ...

You create your pages according to 'Templates', which can be seen as 'Recipes' for your pages. You have to have at least One Template in your system before you can start creating Pages. Every Page is based upon a Template, and no page can exist without its Template ...

Click 'Publishing->Templates ...' now!

If you click 'Dashboard' at the root of your menu you will return back here. To access the root of your menu, click the left arrow at the top of your Sliding Menu ...

PS!Click the Edit links to view any specific templates ...

Every Template contains a bunch of WebPart Templates. These WebPart Templates can be positioned exactly as you wish on your page. When editing a Template, use the Arrow Buttons to position your WebParts ...

Go check out 'Publishing->Templates ...' one more time, and see how you can move stuff around on your 'surface' by clicking the Arrow Buttons ...

When positioning your WebPart Templates, realize that you're not really 'positioning' them, but rather you are changing their width, height and margins. In the beginning this might feel a little bit cumbersome, though after some time you'll hopefully appreciate this 'floating layout' and become used to it ...

Realize also that especially the bottom and right margins might create funny looking WebParts since they're not really visible while editing. If you're having weird results, make sure your right and bottom margins are 0 by double clicking them, which should set them back to zero ... ;)

Double clicking any of the arrows will either maximize or minimize their associated property ...

A WebPage Template must contain at least one WebPart Template. A WebPart Template is also a 'Type Definition' for your WebParts. WebPart Templates have names such as 'Content' and 'Header', which are publishing modules for showing large letters and rich text fragments. Every WebPart Template is based upon one plugin type.

Meaning if you have one page, based upon a WebPage Template, with 5 WebPart Templates, you'll have a WebPage with 5 WebParts where each WebPart can be different 'Applications'. This might be any combination of Applications, such as Text Fragments, Headers, CRUD Forms and such ...

However, we'll stick to 'Publishing' as we promised in the beginning, and focus on the Publishing Modules ...

If you edit the default Template created by the system for you, you can see how it has Menu, Header and Content as 'Module Names'

Go check it out while I hang around here ...

The Menu is similar to the Sliding Menu to the left which you're using yourself, while the Header will show an H1 HTML element [Header Element] and the Content module will show Rich Editable Content.

If you have more types of Modules in your installation of Magix, these might also show up as selections in the DropDownBoxes visible while editing your Templates ...

No go to Templates and change the Name of Template 'M+H+C'. Change it to 'Testing'. This can be done by clicking directly on the text where it says 'M+H+C'. Then open up 'Pages ...' and click your root page.

Do you see how the name of the module in the DropDown box, roughly at the middle of the screen has changed now to 'Testing'. This is because that DropDown box is being used to select a Template for your WebPage.

Then try to change the Name of your Templates by clicking e.g. Header while editing your Template, and type in 'Header2'. Now edit your Page and see how this change reflects from the Template and to the Page.

Go take a look, while I chill ...

Important: If you change the number of WebParts in your Template, or you change the type of Module of your WebPart Template, then all pages built upon that Template will have to be resaved, and you'll probably loose data.

It is therefor important that you create your templates first, and then don't edit these two properties while they're already in 'Production' ...

Try to Create several different Templates, and have slightly different values for their width, height, margins and such.

Be certain of that you've added different widths of your Menu Containers and different Left Margins

Make sure they've got the same type of modules in the same container

Then use these different Templates for different pages which you create in your Pages hierarchy

If you now access the root of your website, and try to browse around by clicking different buttons, you can see how the WebPart Containers are 'jumping around' on the screen ...

Init. of WebParts

WebParts initializes differently depending upon where you're coming from, and from which Template type you're coming from dependent upon to which Template type you're entering. And also according to which type of WebParts, or Modules they are ...

For instance the sliding menu will not reload as long as the container it is within on one page template, is the same container it is within on the next page template ...

Too confusing ...?

Just remember; always have a Menu positioned in the same container [first one for instance?] on all of your templates, unless you really know what you're doing ...

Play around with the system by creating some new Templates, copying them, change their weparts type between Header, Content and SliderMenu. Then when you come back, we'll start diving into Applications ... ;)

Applications ...

Most applications will be WebParts. This means that you can inject them into any WebPart onto any WebPage you wish.

Let's create an Application ... :)

First make sure you have one Template in use in one of your pages which has one WebPart Template with the Module Type of 'MetaView_Single' ...

Then click on 'MetaTypes/Meta Views' ...

Create a new View called 'CollectEmails' ...

Add three properties to your form, name them

\tName\tEmail\tSubscribe

Change their description to something meaningful ...

Make sure you change the 'Type Name' of your object to 'EmailSubscription'

Attach two Actions to your 'Save' property.

\tMagix.DynamicEvent.SaveActiveForm\tMagix.DynamicEvent.EmptyActiveForm

The first Action will save your form, while the second one will empty it.

Now try to View your form in preview mode, and test it out by typing in your email and name, and clicking Submit to save your Object ...

PS!Obviously it's crucial that the 'Save' action runs before the 'Empty' action, in case you wondered ... ;)

TypeNames ...

If you take a look at your Meta Objects now you will see a new object with the Type Name of 'EmailSubscription'. The 'Type Name' property from your MetaView decides the Type Name of your Objects ...

These 'Type Names' are important to distinguish from different types of Objects.

One Object might be of type 'Customer', while another object might be of type 'Email', and so on. What names you give your Types is crucial! Name clashes here might create very hard to track down bugs and such ...

Take some care when naming your Types!

It's probably a good practice to some how make sure they've got unique names, also across your organization if you want to use plugins made by others.

We encourage people to use type names such as; "CompanyName.Department.Customer", and never 'Customer' directly. In fact, your homework for this lesson is to go and rename your 'Customer' TypeName, and rename the Type Name to; 'CompanyName.Department.Customer'. Where Company Name and Department are your company name and your department ...

OpenID

Did you know that you can use Magix as both a Relying Party and OpenID Provider?

[Read more about Open ID here ...]

If you need an OpenID token to log into some website somewhere, then you can append ?openID=admin after the root URL to your website, if the admin user is the User you'd like to log in with.

[Full example; yourdomain.com/?openID=admin]

This will redirect from the website you're trying to log into, and back to your website, which in turn will ask you for admin's password.

Once successfully logged into your own Magix website, your website will redirect back to the website you're trying to log into, and tell it that it 'has prof of that you are who you claimed you were'.

You can also associate other OpenID tokens, such as your Yahoo account or Blogger account, with your Magix User internally. This will allow you to log into Magix with that OpenID Account.

If your blogger account is 'magix' for instance, then your OpenID 'username' [claim] would become 'magix.blogspot.com'

Talk to CEO

If you make something cool with Magix, that you want to share, then we'd love to get to know about it. It can be an instructional YouTube video about how to Get Started, Tips and Tricks etc. It can be a Tutorial Blog you've written about how to install Magix on your server. It can be a book you have written about the O2 Architecture. It can be an Action, or a collection of Actions, which you think is awesome. Anything really!

As long as it has Value for our Community somehow, we'd love to know about it, so that we could help you promote it, and more importantly; Helping our Community out ... :)

Our CEO's Name and Email address is; Lissa Millspaugh - lissa.millspaugh@winergyinc.com

Our CTO's Name and Email address is; Thomas Hansen - thomas.hansen@winergyinc.com

If your stuff is of 'geeky nature', it's probably best to send it to our CTO, or at least CC him in ... ;)

2. Meta Actions

Meta Actions, or 'Actions' for short, are dynamically created actions within the system, either created by the system itself during installation or something, or Actions createdby users, such as you.

These Actions can then later be invoked by some sort of event, for instance a user clicking a button, or something similar

Basically, if it can be described as a 'verb', it's probably an Action ... ;)

SaveActiveForm

Magix.DynamicEvent.SaveActiveForm

Magix.MetaView.CreateSingleViewMetaObject

Will save the currently active Single-View Form.Will determine which form raised the event originally, and explicitly save the field valuesfrom that Form into a new Meta Object with the TypeName from the View ...

ScrollClientToTop

Magix.DynamicEvent.ScrollClientToTop

Magix.MetaView.ScrollClientToTop

Will scroll the browser all the way to the top. Useful for forms which are longer than the fold, when you submit them, and want to scroll to the top to display some message

LoadObjectIntoForm

Magix.DynamicEvent.LoadObjectIntoForm

Magix.MetaView.LoadObjectIntoForm

Will assume a Meta Object is just loaded correctlyinto the Node structure somehow, and pass it on to any matching TypeName Single Views which will handle it and load up the values for the specific Meta Object into their specific editor controls

SetFocusToFirstTextBox

Magix.DynamicEvent.SetFocusToFirstTextBox

Magix.MetaView.SetFocusToFirstTextBox

Will set focus to the first control on the form if raised from within the same WebPart as the form it's trying to set focu to. Works only with Single Views. Explicitly set the 'OriginalWebPartID' to override to set focus to another WebPart's first TextBox on the Page

ValidateObjectPropertyMandatory

Magix.DynamicEvent.ValidateObjectPropertyMandatory

Magix.Common.ValidateObjectProperty

Will validate the given 'PropertyName' Value of the Active MetaView within the WebPart agains the 'Type' parameter. Will throw an exception if validation fissles with an error message back to user, meaning if you'd like to stop saving of a MetaObject due to validation fissling, you'll need to have this Action before the SaveActiveForm Action. The 'Type' parameter can be either 'email', 'mandatory', 'number', 'full-name' or 'url' - Plus any of the gazillion pluginsyou might have other places in your system. For this Action the 'PropertyName' is 'Name', and the 'Type' is 'mandatory', which just means that if there's no value, or only spaces and non-alphabetical-characters, then it's considered empty, and an exception will be thrown

ValidateObjectPropertyFullName

Magix.DynamicEvent.ValidateObjectPropertyFullName

Magix.Common.ValidateObjectProperty

Will validate the given 'PropertyName' Value of the Active MetaView within the WebPart agains the 'Type' parameter. Will throw an exception if validation fissles with an error message back to user, meaning if you'd like to stop saving of a MetaObject due to validation fissling, you'll need to have this Action before the SaveActiveForm Action. The 'Type' parameter can be either 'email', 'mandatory', 'number', 'full-name' or 'url' - Plus any of the gazillion pluginsyou might have other places in your system. For this Action the 'PropertyName' is 'Name', and the 'Type' is 'full-name', which means that at least two names need to be given, and the names will be normalized such as 'Hansen, Thomas' with the last name(s) first and all names automatically capitalized as a function of this Action. No automagic Capitalization will occur if Capital letters are found in the middle of a string though, such as 'McAngus'

ValidateObjectPropertyEmail

Magix.DynamicEvent.ValidateObjectPropertyEmail

Magix.Common.ValidateObjectProperty

Will validate the given 'PropertyName' Value of the Active MetaView within the WebPart agains the 'Type' parameter. Will throw an exception if validation fissles with an error message back to user, meaning if you'd like to stop saving of a MetaObject due to validation fissling, you'll need to have this Action before the SaveActiveForm Action. The 'Type' parameter can be either 'email', 'mandatory', 'number', 'full-name' or 'url' - Plus any of the gazillion pluginsyou might have other places in your system. For this Action the 'PropertyName' is 'Email', and the 'Type' is 'email'. Please note that if you wish to allow for _either_ a valid email address, _or_ an empty value, you need to pass in 'AcceptNull' being true

ValidateObjectPropertyNumber

Magix.DynamicEvent.ValidateObjectPropertyNumber

Magix.Common.ValidateObjectProperty

Will validate the given 'PropertyName' Value of the Active MetaView within the WebPart agains the 'Type' parameter. Will throw an exception if validation fissles with an error message back to user, meaning if you'd like to stop saving of a MetaObject due to validation fissling, you'll need to have this Action before the SaveActiveForm Action. The 'Type' parameter can be either 'email', 'mandatory', 'number', 'full-name' or 'url' - Plus any of the gazillion pluginsyou might have other places in your system. For this Action the 'PropertyName' is 'Age', and the 'Type' is 'number'. Please note that if you wish to allow for _either_ a valid number address, _or_ an empty value, you need to pass in 'AcceptNull' being true. The valid characters in a 'number' are; '0123456789,. '

CreateGallery

Magix.DynamicEvent.CreateGallery

Magix.Common.CreateGallery

Will create a Gallery object from the given 'Files' list within the 'Folder'

IncludeCSSFile

Magix.DynamicEvent.IncludeCSSFile

Magix.Core.AddCustomCssFile

Will include a CSS file onto the page, even in an Ajax Callback if you wish. Change the 'CSSFile' parameter to choose which CSS file you'd like to include

CreateQRCode

Magix.DynamicEvent.CreateQRCode

Magix.QRCodes.CreateQRCode

Will create a QR Code with the given 'FileName' path and filename, which should end with .png. The QR Code will point to the given 'URL', and it will use the textures of 'BGImage' and 'FGImage' to render the code. The QR Code will have 'RoundedCorners' radius of rounded corners, and it will be 'AntiPixelated', and have the descriptive text of 'Text'. The QR Code will use 'Scale' number of pixels per square to render

PlaySound

Magix.DynamicEvent.PlaySound

Magix.Core.PlaySound

Will play the given 'File' sound. If the sample sound file doesn't work, you've probably got something wrong with the setup of your Web Server. Make sure the file extension .oga is associated with the MIME type of audio/ogg. PS! For the record; Cool-Breeze.oga and The-Last-Barfly.ogg are both songs composed by our CTO Thomas Hansen. They are both performed by Thomas Hansen and his wife Inger Hoeoeg, and are to be considered licensed to you under the terms of Creative Commons Attribution-ShareAlike 3.0

PauseSound

Magix.DynamicEvent.PauseSound

Magix.Core.PauseSound

Stops Any sound or music currently being played

ResumeSound

Magix.DynamicEvent.ResumeSound

Magix.Core.ResumeSound

Resumes any sound or music previously being halted through 'PauseSound' or other similar mechanisms. PS! Will throw exception if no sounds have been played, and hence no resuming can occur in any ways

EmptyActiveForm

Magix.DynamicEvent.EmptyActiveForm

Magix.MetaView.EmptyForm

Will empty the currrently active Editable Form. Will determine which form raised the event originally, and explicitly empty that form only. Useful for things such as 'Clear Buttons' and such ...

RedirectClient

Magix.DynamicEvent.RedirectClient

Magix.Common.RedirectClient

Will redirect the client's browser to thegiven URL parameter

ImportCSVFile

Magix.DynamicEvent.ImportCSVFile

Magix.Common.ImportCSVFile

Will import the given 'FileName' from the given 'Folder' and transform to MetaObjects using the given 'MetaViewName'

ShowDefaultMessage

Magix.DynamicEvent.ShowDefaultMessage

Magix.Core.ShowMessage

Will show a 'Message Box' to the User. If you add 'IsError' to true, the message box will be in error mode, meaning red probably, signifying a an error

TurnOnDebugging

Magix.DynamicEvent.TurnOnDebugging

Magix.Common.SetSessionVariable

Will turn on 'Debugging', meaning you'll have a wire-gridcovering your screen to see the 40x18 pixel 'grid-lock', plus you'll also get to see every single Action ever raised on the server shown in an 'Action Stack Trace' Window. This only affects your session, meaning it should be safe to do in production to track down errors and such in live software ...

TurnOffDebugging

Magix.DynamicEvent.TurnOffDebugging

Magix.Common.SetSessionVariable

Will turn _OFF_ 'Debugging', meaning you'll no longer have a wire-grid covering your screen, plus the stack tracing of actions on the server will disappear. Only affects your session, and no other logged on users ability to see debugging information ...

ViewMetaViewMultiMode

Magix.DynamicEvent.ViewMetaViewMultiMode

Magix.MetaView.ViewMetaViewMultiMode

Will load a grid of all Meta Objects of type already loadedin current activating WebPart. If you want to load a specific type, then you can override the type being loaded by adding 'MetaViewTypeName' as a parameter, containing the name of the view

SendEmail

Magix.DynamicEvent.SendEmail

Magix.Common.SendEmail

Will send an email to the 'To' address

SendEmailFromForm

Magix.DynamicEvent.SendEmailFromForm

Magix.Common.SendEmailFromForm

Helper to further assist on creating 'Send Email Forms'. Will send an email with the given 'Body' and 'Header' to the email address within the 'Email' field of the Active SingleView Form View.It will also replace every instance of [x] with the MetaView's property with the same named value. Meaning if you've got a MetaView property called 'Name' it will replace all occurances of [Name] with the value from the MetaObject's property of 'Name' in both the body and the header.PS! This Action can be seen in 'Action' in the 'Magix.Demo.SendEmail' MetaView

ReplaceStringValue

Magix.DynamicEvent.ReplaceStringValue

Magix.Common.ReplaceStringValue

Will transform every entity of 'OldString' found in 'Source' into the contents of 'NewString' and return as a 'Result', output node. You can also use 'SourceNode', 'NewStringNode' and 'ResultNode' to override which Nodes to read values and store result into

MultiAction

Magix.DynamicEvent.MultiAction

Magix.Common.MultiAction

Will raise several Actions consecutively, in the order they're definedin the 'Actions' node. Each Action needs a 'Name' and its own set of parameters through its 'Params' node.All 'Params' nodes will be copied into the root node before every event is raised. This means that yourRoot node will become VERY large after subsequent actions. Be warned ...

GetSingleMetaObject

Magix.DynamicEvent.GetSingleMetaObject

Magix.Common.GetSingleMetaObject

Will put every property from the given Meta Object, into the given Node, with the name/value pair as the node name/value parts, assuming they're all strings. Copy this Action, and make sure you _CHANGE_ its IDtowards pointing to the ID of a real existing Meta Object, to fill in values from one ofyour Meta Objects into a Node, maybe before sending the node into another even, usingMultiActions or something ...

SetMetaObjectValue

Magix.DynamicEvent.SetMetaObjectValue

Magix.MetaType.SetMetaObjectValue

Will set the value of the given MetaObject [ID] to the Value of your 'Value' node at the 'Name' property.

SerializeActiveSingleViewForm

Magix.DynamicEvent.SerializeActiveSingleViewForm

Magix.MetaView.SerializeSingleViewForm

Will find the Form that raised the current eventchain, and query it to put all its data flat out into the current Node structurewith the Name/Value as the Node Name/Value pair. Note; Can mostly only be raised from within a Single View MetaView, unless you're doing other overrides

LoadSignatureModule

Magix.DynamicEvent.LoadSignatureModule

Magix.Common.LoadSignatureForCurrentMetaObject

Will load the Signature Module inwhatever container its being raised from. And set the 'Value' property of the givenMetaObject Column Property Name to the signature signed on the Signature module ...

ExportMetaView2CSV

Magix.DynamicEvent.ExportMetaView2CSV

Magix.Common.ExportMetaView2CSV

Will render the 'Currently Viewed' MetaView intoa CSV file [Microsoft Excel or Apple Numbers etc] and redirect the users client [Web Browser] to the newly rendered CSV file. 'Currently Viewed' meaning the view that contained the control that initiated this Action somehow. PS! If you explicitly create a 'MetaViewName' parameter, and set its name to another MetaView, then that MetaView will be rendered instead

ViewCarsInPopup

Magix.Demo.ViewCarsInPopup

Magix.MetaView.ViewMetaViewMultiMode

Will load up any Magix.Demo.Car objects into a MultiView form, within a popup window, using the Magix.Demo.ImportCars MetaView. If there are no items you can run the 'Magix.DynamicEvent.ImportCSVFile' action, which by default will point towards a CSV file which contains 'Cars' data

CreatePDF-Book

Debug-Copy-Magix.PDF.CreatePDF-Book

Magix.PDF.CreatePDF-Book

3. ImageGallery

[ActiveModule]

Magix.Brix.Components.ActiveModules.PublisherImage.ImageGallery

Description:
Level1: Represents an Image Gallery plugin for the Publishing system. Allows for browsing of Images through a Gallery lookalike UI

Basetypes
Magix.Brix.Loader.ActiveModule

Methods

public override void InitialLoading(Node node)

Level3: You _Must_ call this one, if you override it, to set the Module's DataSource property. And in order to get some other types of wiring running correctly for the module

Properties

public string GalleryName

Level1: The Unique name of the Image Gallery