



ENOVIA DesignSync Enterprise DesignSync Administration User's Guide

3DEXPERIENCE 2022

©2021 Dassault Systèmes. All rights reserved. 3DEXPERIENCE®, the Compass icon, the 3DS logo, CATIA, SOLIDWORKS, ENOVIA, DELMIA, SIMULIA, GEOVIA, EXALEAD, 3D VIA, BIOVIA, NETVIBES, IFWE and 3DEXCITE are commercial trademarks or registered trademarks of Dassault Systèmes, a French "société européenne" (Versailles Commercial Register # B 322 306 440), or its subsidiaries in the United States and/or other countries. All other trademarks are owned by their respective owners. Use of any Dassault Systèmes or its subsidiaries trademarks is subject to their express written approval.

**DASSAULT
SYSTEMES**

Table of Contents

Getting Started	1
Overview of Enterprise DesignSync Administration	1
Introduction to EDA Terminology	1
Prerequisites	3
Enterprise Design Synchronization	3
Managing DesignSync Developments.....	5
Getting Started Managing DesignSync Developments	5
Add Development	5
Adding a Development.....	5
Associating Sub-Properties with Developments.....	10
View/Edit Developments	11
Using this Panel	11
Field Descriptions.....	11
Related Topics	14
Edit Development	14
View Development.....	14
Related Topics	15
View Development Areas	15
Field Descriptions.....	15
Select Shared Tool Suite	16
Development Settings.....	17
Add Development Settings	18

- Field Descriptions..... 18
- Edit Development Settings 21
 - Field Descriptions..... 22
- View Development Settings..... 23
 - Related Topics 23
- Understanding Tool Suites/Shared Tool Suites 23
 - Related Topics 24
- Add Tool Suite/Add Shared Tool Suite 24
 - Field Descriptions..... 25
- Edit Tool Suite/View or Edit Shared Tool Suite..... 28
 - Field Descriptions..... 29
 - Related Topics 32
- View Tool Suite/View Shared Tool Suites..... 32
 - Related Topics 32
- Understanding Tools/Shared Tools 33
 - Related Topics 33
- Add Tool/Add Shared Tool 33
 - Field Descriptions..... 34
 - Related Topics 35
- Edit Tool/Edit Shared Tool..... 36
 - Field Descriptions..... 36
 - Related Topics 37
- View Tool..... 37

Enterprise DesignSync Administration User's Guide

- Development Assignments 37
 - Add Development Assignment 38
 - Field Descriptions..... 39
 - Edit Development Assignment..... 41
 - Field Descriptions..... 42
 - View Development Assignment 43
 - Related Topics 43
- Managing Development Areas 45
 - Setting up Development Areas 45
 - Prerequisites 45
 - Getting Started Using Development Areas 45
 - Related Topics 46
- List Areas..... 46
 - Explanation of Fields..... 47
 - Related Topics 48
- Make Area 48
 - Field Descriptions..... 50
 - Related Topics 51
- Remove Area..... 52
 - Field Descriptions..... 53
 - Related Topics 53
- Join Area 53
 - Field Descriptions..... 54

- Related Topics 54
- List Developments 55
 - Explanation of Fields..... 55
 - Related Topics 56
- Change Area 56
 - Field Descriptions..... 58
 - Related Topics 59
- Console 59
- Accessing a SyncServer with User Authentication 59
 - Username 60
 - Password 60
 - Save password..... 60
- TroubleShooting 61
 - No Default User Specified 61
- Index 63

Getting Started

Overview of Enterprise DesignSync Administration

Enterprise DesignSync Administration defines a comprehensive environment for managing the design data, methodology, tools, and assignments for users working on a common development project. The definition of the development is maintained on DesignSync servers and accessed via the DesignSync Web User Interface. End users have a simple client interface to create an area to work on the design data and invoke pre-configured tools, all ensuring a common methodology is employed by all users

The application interface to EDA consists of two primary components. There is a Web UI intended primarily for use by project administrators and/or project leaders. Using this UI, the administrators define the design data to use, the tools, and the methodology that all the development's users adopt and apply. This interface is described in [Getting Started Managing DesignSync Developments](#).

End users have a simple client interface, available in either GUI or command-line format, to create an area to work with the design data and invoke pre-configured tools, ensuring a common methodology is employed by all users. The GUI interface is described in [Setting up DesignSync Areas](#). The Command-line interface is described in the *ENOVIA Synchronicity Command Reference*: [sda](#).

Note: The ability to configure, maintain and deploy Enterprise DesignSync Administration is an administration feature.

Introduction to EDA Terminology

Development

A collection of all the information that defines a development project. This includes information on the data being developed, the users operating on the data, the tools being used and their setup information, and the DesignSync servers and physical paths where development is taking place.

Development Instance

The manifestation of a development on the local file system associated with a DesignSync server. This means that all of the design data defined for the development is populated into a set of automatically updated areas that EDA manages. The module defining a development's settings is also fetched and managed.

Development Setting (or Setting)

A module which manages a set of files that defines how DesignSync clients (and other third party, tools) should operate when working on data for a specific development. For example, this may include whether a locking model is in effect, or additional TCL files are sourced to define/modify commands.

Development Assignment (or Assignment)

A set of attributes which describe the type of access that a specific user is to have when working on the development's design data in their development area. The attributes can define a subset of the design data and tools to use. Often the assignment is a functional set of work, for example, you might have a release engineer, developer, tester, and writer set of assignments viewing the same data, but not in the same way. , The writer assignment might filter out code, the developer assignment might filter out the documentation, the release engineer assignment might point to a version tagged for release, etc.

Development Server

A standard DesignSync server which hosts the definition of Developments and manages the data associated with Development Instances. The Development Server must be configured as a Mirror Administration Server (MAS).

Development Area (or Area)

A user's workspace in which they invoke tools which are configured to specifically run against a development's design data. The methodology of the tools run by the user is defined by the development's settings.

DRR (or Data Replication Root)

The DRR is part of DesignSync's standard automated data replication system (aka replicate). The DRR is a set of local workspaces that capture the dynamically changing data and the static unchanging data of a DesignSync URL. If the URL refers to a DesignSync module, then an entire module hierarchy is managed by the DRR, including the creation and auto-updating of the module cache. A single, development-specific file cache is also created and maintained so that only a single version of each file is kept for the development. All other uses of the file are stored as hardlinks into the file cache. For more information on data replication, see the *ENOVIA Synchronicity DesignSync Administrator's Guide*: [Introduction to Data Replication](#).

sda (DesignSync Development Area Management Tool)

The sda tool is a DesignSync client application that manages an end user's experience with EDA. It sets up work areas and launches tools to run on the design data in the area. There is a command line interface to sda and also a graphical interface. The command interface prompts the user for all needed options.

Template Development Setting (or Template)

EDA provides a set of predefined development setting modules that define a methodology for Analog design, Digital design, and Software design. The template setting modules are installed on every DesignSync server and accessible in a read-only manner.

Tool Suite

A collection of tools/applications that can be defined for an EDA development and setup to run in a pre-configured way.

Tool

An individual tool/application that can be invoked by an end user to run on a development's design data within a user's development area. The tool can be configured by the development setting, or the owning tool suite, or even the tool's definition itself.

Prerequisites

In order to be able to access any of the DesignSync Web User Interface panels for EDA, the DesignSync server must be configured first as a Mirror Administration Server (MAS). This is done via the Mirror->General Settings panel. For more information, see the *ENOVIA Synchronicity DesignSync Administrator's Guide*: [Setting Up a Mirror Server](#).

A server that manages the design data being used by a DesignSync development must be set-up as a Mirror Repository Server (RS). This is because EDA will be automatically creating mirrors to update the locally fetched design data associated with the development. Likewise, a server that manages the development's setting module will also need to be running as a Repository Server. For more information, see the *ENOVIA Synchronicity DesignSync Administrator's Guide*: [Setting Up a Mirror Server](#).

Any user who is creating or maintaining DesignSync Developments must have the access rights to use the DesignSync Development and any specific commands and menus desired. For more information setting up access permissions for DesignSync Developments, see [Access Controls for DesignSync Developments](#).

Enterprise Design Synchronization

DesignSync features automatic Enterprise Design Synchronization for maintaining the up-to-date information across your enterprise system. When objects which are being represented in the Enterprise System are created, modified, or removed in the

DesignSync system, those changes can be pushed to the Enterprise Server, updating the hierarchical structure in that system and the links back to the DesignSync objects.

The push can occur either automatically using a timer mechanism, or manually. The automatic synchronization mechanism is configurable through the DesignSync Administrator. Any transactions queued when the timer interval is reached are then pushed to the Enterprise server. For more information about configuring automatic synchronization, see the *DesignSync Administrator's Guide*: [Site Options](#).

When the following DesignSync operations are performed, a corresponding entry to update the Enterprise Product structures is placed in the synchronization queue:

- Tag was removed (tag)
- Tag was modified (tag)
- Hierarchical reference was added (addhref)
- Hierarchical reference was removed (rmhref)
- Hierarchical reference was modified (edithref)

Note: New tags do not automatically update in the ENOVIA system. You can create your own triggers to create new Enterprise Design Objects or modify existing Enterprise Design Objects when tags are added.

DesignSync also synchronizes product information, such as Product Type or Policy information, which can be entered either on the ENOVIA system or within DesignSync using the command line `entobj` commands. For more information, see the *Synchronicity Command Reference*: [entobj](#) commands.

Entries in the synchronization queue are processed in the order in which they were received. You can view the queue in the DesignSync WebUI. For more information about viewing or selectively processing entries in the synchronization queue, see *DesignSync Administrator's Guide*: [Enterprise Design Synchronization Queue](#).

Managing DesignSync Developments

Getting Started Managing DesignSync Developments

DesignSync developments are managed through the DesignSync Web UI.

To access the Web UI, enter the hostname and port of the DesignSync server as a URL into a supported web browser.

```
http://<host>:<port>
```

You may be prompted for a username/password.

From the DesignSync Web UI, you can access DesignSync Developments.

The primary functionality is available through the [Add Development](#) and [View/Edit Development](#) links.

For more information on using the DesignSync web UI, see the [ENOVIA Synchronicity DesignSync Data Manager Administrator's Guide](#).

Add Development

When adding developments, you can define the properties specific to the development and use the sub-tables to associate the development with the necessary sub-components to provide full functionality.

Adding a Development

Click on the image for more information about each field.

DesignSync Developments | Add Development

Complete this form to specify the details for the development.
You can also add, edit or delete associated settings, assignments and tool suites.

*** Denotes a required field**

Name*	ChipNZ215
State	Created
URL*	sync://ABCco.com:30126/Modules/Chip/NZ214
Selector*	R17:Gold
Development path*	/home/frsmith/developments
Development Area root	<input checked="" type="radio"/> Default path <input type="radio"/> Custom path: <input type="text"/>
Allow user-defined development area paths	<input type="checkbox"/>
UNIX Group	users
Data replication root	<input checked="" type="radio"/> Default path <input type="radio"/> Custom path: <input type="text"/>
Scrub Period	7
Description	Chip NZ215 Development environment
Allow external development areas	<input type="checkbox"/>
External development area path	<input type="text"/>

Field Descriptions

This section describes the fields available on this panel.

Name

Enter the DesignSync development name. The name of the development must be unique with respect to all other developments that are defined on the DesignSync server.

Note: For optimal function, you should avoid using any [reserved characters](#) in the name.

This is a required field.

State

Select the state value.

- Created - Only the metadata for the development is saved, allowing you to further [modify the properties](#) for the DesignSync development. If the development is only being created, not enabled, you are not required to fill in all the properties for the development. The development can be enabled later using by specifying **Enable** from the View/Edit Development page. If you leave the

development in the Created state, it can provide a template for creating subsequent developments.

- **Enabled** - All of the properties for the development are saved and validation is performed immediately. If needed, the system creates the physical directories, and, if necessary, module or vault indicated by the [URL](#), and initializes the appropriate DRR and mirrors for the development. Upon successful enabling, the DesignSync development is immediately available for use. In order for the development to be enabled, it must have at least one Development Setting and at least one Tool Suite of the DesignSync type. If these conditions aren't met, the development definition may continue to be defined or the "enable" operation canceled – which saves the development, but leaves it in the created state, allowing you to add any missing components required at a later time by editing the development.

This is a required field.

URL

Enter the SyncURL for the the design data to which the EDA development will be using. This might typically be the URL to a modern module with a hierarchy of submodules beneath it. However, it can also point to a URL of data managed in a files-based DesignSync server vault. This is a required field.

If the [State](#) is Enabled, validation is performed on the URL to make sure the managed vaults exist. If the module or vault does not exist, DesignSync automatically creates it. The newly created vault or module is owned by the default user for the mirror system, and, if it is a module, created with an initial comment of, "Automatically created by the EDA system." If it cannot be created, the Development is not enabled, but remains in a created state and the user receives an appropriate error.

Selector

A valid selector. You should distinguish between branch and version selectors. If you are specifying a branch other than Trunk, specify both the branch and version as follows:

'<branchtag>:<versiontag>', for example,

```
'Rel2:Latest'
```

Development path

Path to the development's storage directory accessible by the server to store ancillary development data, including a copy of the development setting used to control the development. In addition, this development path, by default, contains the location where all the design data is fetched and managed locally into the development's DRR. By

default, all of the development areas defined by users are placed into this development path as well. There are optional ways to override the location for the DRR and development areas.

Tip: Create a dedicated development storage directory on the server to store all the development hosted on that server, for example:

```
/developments
```

The development path would then something unique, but memorable, like

```
/developments/<DevName>
```

Development Area root

Defines the relative path to the directory that will hold all the users' development areas created for this development.

- Default path – Specifies to place all users' development areas under the “Development path” (defined above). EDA creates the user's specific development area in the subdirectory:
`DevelopmentAreas/<userName>/<developmentAreaName>`
- Custom path - Specifies a relative path to the directory that will hold all the users' development areas created for this development. This path may use environment variables to build the path name. This custom path is relative to the “Development path” (defined above).
For example, you might set up a directory structure like this, using the “\$USER” variable.
`DevAreas/$USER`

Note: On Windows systems, still use \$USER and this will automatically be converted to the USERNAME value

Individual development area directories are created at either the default path or custom path as defined above. The design data referred to by the development is then populated into a sub-directory named after the leaf of the data URL.

So, for example: if the Development Path for a project is `/developments/Chip`, and the custom development area path is `DevArea/$USER`, and the data URL is `sync://<hostname:<portnumber>/Modules/MyDev/top`, and user “fred” creates a development area named “Work1”, then the complete path to fred's development area data is:

```
/developments/Chip/DevArea/fred/Work1/top
```

Allow user-defined development area paths

Indicates whether the users can specify their own path name for their development areas, or are required to use the default or custom path specified by the [Development Area root](#) option.

UNIX Group

Sets the UNIX group associated with the development folder on the server. When enabling a development, DesignSync creates a development folder on the server with the s-bit set and can assign it to the specified group.. By default this field is blank. The drop-down lists all groups associated with the owner of the DesignSync installation.

Note: The specific permissions set on the top-level development folder are 1750 (Sticky bit, READ/WRITE/EXECUTE by user, READ/EXECUTE by group, no permissions for world).The specific permissions set on the individual development definition (sub)folders are 1770 (sticky bit, READ/WRITE/EXECUTE by user, READ/WRITE/EXECUTE by group, no permissions for world)

Data replication root

Path to the Data Replication Root (DRR) which holds the automatically populated design data for the development. This DRR splits the fetched design data into a dynamic area (useful as mirror directories), a static area for defining a module cache, and a file cache which holds all the necessary versions of the individual files used by the development.

- Default path - Uses the existing, or creates a new data replication root in the ../DesignSync/DRR subdirectory of the [Development path](#).
- Custom path - Specifies a common path name that multiple developments use as a common data replication root. This is valuable if two or more developments share common data. Rather than have each development contain its own DRR (and thus its own copy of all the populated design data), a common DRR path can be specified so that common files are replicated only once for all the developments using the DRR. This path may use environment variables to build the path name.

For more information on data replication, see the *ENOVIA Synchronicity DesignSync Administrator's Guide*: [Introduction to Data Replication](#).

Scrub period

Period in days for cleaning of the unused data in the file and module caches maintained by the data replication system for the development. For more information on data replication cache scrubbing, see the *ENOVIA Synchronicity DesignSync Administrator's Guide*: [Cleaning and Removing Data Replication](#).

By default, this is set to zero, meaning that scrubbing is disabled. Typical data replication setup uses a scrub time of 7 days.

Note: The scrubbing process only uses development areas and the DRR available to a development instance that is local to the development server. Any external development instances, associated with external areas, cannot be scrubbed automatically by the system.

Description

Description of the development.

Allow external development areas

Select this option to allow users to create development areas on clients that do not have direct access to the main project path. For example, Windows users, or users on a different network that can access the DesignSync Server but not the physical project path.

External development area path

Path to the directory that will hold external development areas. EDA will also manage the external development instance in this directory. This path may use environment variables to build the path name.

Associating Sub-Properties with Developments

Developments require Development Settings, Tool Suites, and Development Assignments in order to be used. Any defined objects available to the Development appear in the sub-tables underneath the Development definition. From the sub-tables, you can create, select, or delete an object that can be associated with the development.

Note: On the Tool Suites sub-table, you can choose whether to use a Shared Tool Suite or a Tool Suite exclusively available to your development. When you select [Use Shared Tool](#) suite, you launch a filtered list containing only shared Tool Suites.

Related Topics

[View/Edit Developments](#)

[Add Development Settings](#)

[Add Tool Suite/Add Shared Tool Suite](#)

[Add Tool/Add Shared Tool](#)

[Add Development Assignment](#)

View/Edit Developments

This page contains the View/Edit Development page and information about how to use it.

Using this Panel

Select the development by clicking the checkbox next to the development name. You can then perform several operations with the development using any of the buttons at the bottom of the table. For more information on any of these actions, click the appropriate location in the image below.

Note: The **Default** column of the Development Settings sub-table allows specifying the default setting, which is used by an assignment when there is no specific setting chosen. One and only one setting can be selected as **Default**.

Click on the image for more information about each field.

DesignSync Developments | View/Edit Developments

<input type="checkbox"/>	Name	Edit	Description	Host	State	URL	Development path	Development Area root
<input type="checkbox"/>	ChipNZ215		Chip NZ215 Development environment	Local	<input checked="" type="radio"/>	sync://ABCco.com:30126/Modules/Chip/NZ214	/home/rsmith/developments	
<input type="checkbox"/>	NewDev		New Developments	Local	<input checked="" type="radio"/>	sync://ABCco.com:30126/Modules/Chip/NewDev	/home/rsmith/developments	

Field Descriptions

This section describes the fields available on this panel.

Name

Displays the development name. The Name is an active link that launches the [View Development](#) panel.

Note: You do not need to click the checkbox next to the development name to launch the View Development panel.

Edit

Select the Edit icon to launch the [Edit Development](#) panel which allows you to make any changes to the development definition. If the edits are valid and the development was already enabled before editing, then the edits are applied automatically to the

development instance (may require, for example, repopulating of data). A development still in the created or disabled state only has edits applied to it after being successfully enabled.

Description

Displays the first part characters of the development's description.

Host

Shows the original location of the development. There are two possible values:

- **Enterprise** indicates that the development was originally pushed from an ENOVIA server and therefore is under control of that server.
- **Local** indicates that the development was created on this DesignSync server.

State

Displays the state of the development. Valid values are:

- Created - a blue dot indicates a development in the created state.
- Enabled- a green dot indicates a development in the enabled (or active) state.
- Disabled - a gray dot indicates a development in the disabled (or inactive) state.

URL

Displays the SyncURL for the design data associated with the development.

Development path

Displays the path to the development storage directory on the server to store ancillary data related to the development such as scripts, but also stores the locally populated design data and user development areas.

Development Area root

Displays the relative path name to hold the development areas if different than the default path.

Enable

If the [State](#) is created or disabled, click this button to enable the development. If this is the first time the development is enabled, and the module or vault associated with the development does not exist, DesignSync will create the object, owned by the default user for the mirror system. If the object is a module, it is created with the initial comment of, "Automatically created by the EDA system." If the module or vault cannot be created,

the Development is not enabled, but remains in a created state and the user receives an appropriate error.

Disable

If the [State](#) is enabled, click this button to move the development to the disabled state. A disabled development will no longer be automatically updated by EDA (i.e., the development's mirrors will be disabled). Also, user development areas cannot be created against the disabled development.

View Development Areas

Launches the [View Development Areas](#) page for the selected development. You may only select one development for this operation.

Copy

Launches the [Add Development](#) page with the values for the selected development filled out for the [Development Settings](#), [Tool Suites](#), and [Development Assignments](#). You may only select one development for this operation.

Delete

Deletes the selected development(s).

When a development is deleted:

- The development Instance is deleted from the developments database on the development server.
- The mirror for the development setting is stopped.
- Data replication for the development data stops.

The development instance area and the data within it are not physically deleted. This would typically include all user development areas as well as the development's data replication root holding the mirrored data, module cache, and file cache.

This action does not require a confirmation. The development is deleted immediately.

Note: When a user tries to change (for example, using the [sda cd](#) command), into an area pointing to a removed development, they receive a notice telling them the area has been deleted, but the operation completes successfully.

Help

Opens help to this topic.

Related Topics

[Add Development](#)

[Edit Development](#)

Edit Development

This page contains the Edit Development page and a set of sub-object tables for creating a development.

The fields and sub-object tables displayed are identical to the [Add Development](#) form, except that they are pre-filled with the existing values of the development. You can click the Edit icon in sub-object tables to launch the edit page for that sub-object. You can click Add to create a new sub-object or you can select a sub-object and press Delete to remove it.

Note: Deleting sub-objects does not require a confirmation. You may delete multiple sub-objects of the same type within a single Delete operation.

To save the changes to the development, press the **Submit** button.

To cancel the edit and return to the [View/Edit Development page](#), press **Cancel**.

Related Topics

[View/Edit Developments](#)

[Edit Development Settings](#)

[Edit Tool Suite](#)

[Edit Tool](#)

[Edit Development Assignment](#)

View Development

This page displays the View Development page including the development definitions and the sub-objects associated with the development.

To view the properties associated with a sub-object, click the Name of the sub-object.

To finish viewing the development properties, press the **Ok** button.

To edit the development, press the **Edit** button to open the [Edit Development](#) panel.

Related Topics

[Edit Development](#)

[View/Edit Developments](#)

[View Development Settings](#)

[View Development Areas](#)

[View Tool Suite](#)

[View Tool](#)

[View Development Assignment](#)

View Development Areas

This page shows the development areas associated with the selected development.

You reach this page by selecting the View Development Areas button on the [View/Edit Development](#) page.

Click on the image for more information about each field.

DesignSync Developments
View Development Areas of Development 'ChipNZ214'

Press 'OK' to return to the View/Edit Developments page.

Name	Description	Users	Development Assignment	External Shared	Host	Development path
chipNZ214		FYL	DocWriter	✓	L1015ENO	C:\Users\fyl\developments\DevelopmentAreas\FYL\chipNZ214

Field Descriptions

This section describes the field available on this panel.

Name

Name of the development area.

Description

Description of the development area.

Users

Name of the user who owns the development area as well as the names of users who are joined to the development area, if the area is defined as shared.

Development Assignment

Development Assignment associated with the development.

External

If the development area is external to the development server, this displays a checkmark. Otherwise this field is blank.

Shared

If the development area is defined as a shared development area, this displays a checkmark. Otherwise this field is blank.

Host

Name of the machine on which the development area was originally created.

Development path

Path to the development area on the [Host](#) machine.

OK

Returns to the [View/Edit Developments](#) page.

Help

Displays this help.

Select Shared Tool Suite

This is a filtered list of shared tool suites available for a development. When you select Use Shared on the Tool Suite sub-table when [Adding](#) or [Editing](#) a development brings you to this page where you can select from all the available Shared Tool Suites.

For more information on the Shared Tool Suite, select the name of the Tool Suite.

To add the Shared Tool Suite to the development, select the Shared Tool Suite and press the **Return** button to go back to the Development page.

To cancel without changing the Shared Tool Suites displayed as available on the Development page, press **Cancel**.

To view this help, press **Help**.

Development Settings

Developments in EDA contain one or more settings that typically define the methodology to be used with the development. A setting's data is defined in a separate DesignSync module and is populated into a local copy of a development instance. The type of files normally defined for a setting include:

- DesignSync-based client registry settings stored in ProjectRegistry.reg file
- DesignSync-based custom Tcl scripts, triggers, collection object definitions, and SITaR-related TCL files.
- DesignSync-based dsinit.dss file, used by DesignSync clients for initialization
- DesignSync-based sync_servers.txt file.
- User/Admin-based scripts, initialization files, environmental settings, etc. for any other tools to be used during the development

The directory structure and contents of the settings module is not pre-defined.

When the development is populated, the associated setting module is populated along with it. When a user switches to a development area for a development the environment variable SYNC_PROJECT_CFGDIR is set to point to the directory holding the setting data associated with the development instance.

EDA provides template development settings designed to speed the initial deployment of EDA. You can directly use the template modules or copy and modify them to a new location. The EDA templates are located at:

```
sync://<host>:<port>/Modules/SYNC/DevelopmentSettings/Templates/  
ENOVIA
```

There are 3 different templates: one primarily used for Analog design, one for Digital design, and one for Software design. The differences between each are the way in which the DesignSync registry settings are defined. Each of the templates has an index.html file describing its registry settings in more detail. The template modules can be viewed from the development server's Datasheet page.

Tip: For optimal viewing of the files contained in the template modules, populate the modules and view the HTML files in a web browser.

Add Development Settings

This page contains the Add Development Settings page.

Note: On the [Edit Development](#) page, you can specify the default development setting for a development. The default is used by a development assignment if a specific development setting is not defined for the assignment.

Click on the image for more information about each field.

DesignSync Developments | Add Setting to Development 'ChipNZ215'

Complete this form and press 'Return' to record the changes and return to the Edit Development page, or press 'Cancel' to return without making any changes.

	* Denotes a required field
Name*	<input type="text" value="ChipSettings"/>
URL*	<input type="text" value="sync://ABCco.com:30126/Modules/DevelopmentSettings/ChipSettings"/>
Selector*	<input type="text" value="R17:Gold"/>
Access Mode	<input type="text" value="Reference"/>
Mirror	<input checked="" type="checkbox"/>
Description	<input type="text" value="Gold version of the R17 release settings for Chip Development areas"/>

Field Descriptions

This section describes the field available on this panel.

Name

Enter the name of the development setting. This does need to be identical to the module name. The setting's name must be unique among all settings for the development. Other developments defined on the server may have settings with the same name but different attributes.

Note: For optimal function, you should avoid using any [reserved characters](#) in the name.

This is a required field.

URL

Enter the SyncURL for the DesignSync settings module. This is a required field. The development settings are stored at:

```
sync://<host>:<port>/Modules/SYNC/DevelopmentSettings
```

Note: Validation is performed on the URL to make sure the setting's module exists.

Selector

A valid selector defining which module version of the setting's module should be used for this development. You should distinguish between branch and version selectors. If you are specifying a branch other than Trunk, specify both the branch and version as follows:

'<branchtag>:<versiontag>', for example,

```
'Rel12:Latest'
```

Access Mode

This setting indicates how the development setting module is populated into the development instance. Options are:

- [Make Branch](#)
- [Make Copy](#)
- [Reference](#) (default)

Make Branch

When the development using this setting is enabled, EDA creates a module branch of the originally specified URL@selector. This allows the branched setting module to remain closely associated with the original setting module and thus any changes in the original or the branch can be merged with the other. The new branch is created with an immutable tag value of:

```
<development name>_<setting name>.
```

If this name already exists then the immutable tag is created as:

```
<development name>_<setting name>_n
```

Where *n* is the next available number starting from zero (0).

The new branch is created with a comment of the form “Development Setting module branch created for development <name>”.

The Latest version on that branch will get a tag of *<BranchTag>_Initial*.

After the development is enabled and the setting is branched, the setting URL for this new development is the existing setting module URL and the setting selector will be *<BranchTag>:Latest*

IMPORTANT: The user creating the development must have access to create the new branch for this process to succeed.

Note: You cannot use the Make Branch option on the DesignSync provided Development Setting templates to prevent accidental data change should later releases of DesignSync change the templates.

Make Copy

When the development using this setting is enabled, EDA copies the existing specified setting module URL@selector by populating the files from it, and then checking them in as a new setting module for this development. The new setting module's URL is automatically generated based on the URL specified for the design data for this development: For modules-based development data, the new setting module URL is:

```
sync://<host>:<port>/Modules/SYNC/DevelopmentSettings/Mods/<category>[...]/<module name>[.<setting name>]
```

If the primary design data is files-based, and has a URL of

sync://<host>:<port>/<path> then the new setting module URL will be:

```
sync://<host>:<port>/Modules/SYNC/DevelopmentSettings/Projs/<path>[.<setting _name>]
```

The new module is checked in with an automatically generated comment of the form "Development Setting module generated for development *<name>* by copying *<original setting module version URL>*". The branch is tagged with an immutable tag matching the *<Development Name>[_<Setting Name>]* and the Latest version on that branch is tagged with an immutable tag which is the branch tag with "_Initial" appended. The setting selector for this development will be the *<development name>_<setting name>:Latest* value.

IMPORTANT: The user creating the development must have access to create the new module for this process to succeed.

Reference

Uses the specified setting module URL. (Default) No copying or branching is done.

Mirror

Determines whether the development setting module is to be mirrored. (Default) If it is mirrored, then any time a checkin occurs for the setting module URL@selector, then the setting's module contents are re-populated into the development instance and thus impacts any users with development areas working on the development.

To mirror the development setting module, select the checkbox.

To disable mirroring of the development setting module leave the checkbox blank.

Note: Even if mirroring is disabled, the development server must still be designated as an RS.

Description

Description of the development setting.

Return

Save the changes and return to the previous page.

Cancel

Cancel the changes and return to the previous page.

Help

Launches a web-browser and opens this help page.

Related Topics

[View/Edit Developments](#)

[Add Developments](#)

[Add Tool Suite](#)

[Add Tool](#)

[Add Development Assignment](#)

Edit Development Settings

This page contains the Edit Development Settings page.

Note: On the [Edit Development](#) page, you can specify the default development settings for a development. The default is used by a development assignment if a specific development settings option is not defined for the assignment.

Click on the image for more information about each field.

DesignSync Developments | Edit Setting of Development 'ChipNZ215'

Complete this form and press 'Return' to record the changes and return to the Edit Development page, or press 'Cancel' to return without making any changes.

	* Denotes a required field
Name*	ChipSettings
URL*	<input type="text" value="sync://ABCco.com:30126/Modules/DevelopmentSettings/ChipSettings"/>
Selector*	<input type="text" value="R17:Gold"/>
Access Mode	<input type="text" value="Reference"/>
Mirror	<input checked="" type="checkbox"/>
Description	<input type="text" value="Gold version of the R17 release settings for Chip Development areas"/>

Field Descriptions

This section describes the field available on this panel.

Name

View the DesignSync development settings name. This field is not editable. If the name must be changed, it's easiest to just define a new setting and then delete the old one when finished.

Note: The remaining fields of the Edit Development Setting panel essentially behave the same as the [Add Developments Settings](#) panel.

Related Topics

[View/Edit Developments](#)

[Edit Developments](#)

View Development Settings

This page displays the View Development Settings page. This page is accessible by clicking the name of the Development Settings on the [View/Edit Development](#) page.

To finish viewing the development settings, press the **Ok** button.

To edit the development settings, press the **Ok** to return to the [View/Edit Development](#) page and select the **Edit** icon.

Note: The **Default** column of Development Settings sub-table on the [View/Edit Development](#) page allows specifying the default setting, which is used by an assignment when there is no specific setting chosen. One and only one setting can be selected as **Default**.

Related Topics

[Edit Development](#)

[View/Edit Developments](#)

[View Development](#)

[View Development Areas](#)

[View Tool Suite](#)

[View Tool](#)

[View Development Assignment](#)

Understanding Tool Suites/Shared Tool Suites

A tool suite is set of related tools that can be used for working on a development. Each tool suite is associated with one or more related tools that are available to the users of the development. Tools suites can refer to DesignSync tools, either the current DesignSync version or previous versions, or describe other tools available to the developers.

A tool suite is available to a single development and is considered part of that development.

Shared tool suites are available across developments. This provides the ability to edit a shared tool suite in a single action and update all the developments that use that tool suite. For example, if the DesignSync client is defined as a tool, and you update your

version and want the upgraded version available immediately to all developments, you only need to update the definition in one location.

Tool suites contain [Tool](#) definitions. Shared tool suites contain shared tool definitions.

A tool suite is a component of [Development](#). A shared tool suite is linked to a development.

Related Topics

[Add Tool Suite/Add Shared Tool Suite](#)

[Edit Tool Suite/View or Edit Shared Tool Suite](#)

[View Tool Suite](#)

Add Tool Suite/Add Shared Tool Suite

This page describes the Add Tool Suite and the Add Shared Tools Suite page and the sub-object table for tool and shared tool definitions.

Click on the image for more information about each field.

DesignSync Developments | Add Tool Suite to Development "

Complete this form and press 'Return' to record the changes and return to the Add Development page, or press 'Cancel' to return without making any changes.

* Denotes a required field

Name* EditingTool

Version* x5

Type

DesignSync

Custom type: Documentation Editor

Platform All

Path /usr/contrib/bin/doceditor

Script

Program group Preferred Documentation Editor

Starting directory

Description

Tools						
Display String	Edit	Description	Program name	Arguments	Graphical	Starting directory
DesignSync		DesignSync GUI tool	DesignSync GUI (x64)		✓	
DocEditor		Tool used by the technical writers to create documentation	Documentation Editing Tool		✓	

Add Delete

Return Cancel Help

Note: When working on the Shared Tool Suites, the Submit button replaces the Return button depicted above. The Submit button saves the tool suite definition in the database making it available to all developments using the Shared Tool Suites. Any developments currently using the tool suite may need to be restarted to use the new version;

The Return button, available for Tool Suites that are not shared, updates the Tool Suite by modifying the in-memory definition.

Field Descriptions

This section describes the field available on this panel.

Name

Name of the tool suite. Must be unique within the development. You may use spaces in the name. This tool suite name may be used by users of the development to uniquely identify which tool to run. If you are creating a shared tool suite, the name and version pair must be unique among all shared tool suites and local tool suites.

Note: For optimal function, you should avoid using any [reserved characters](#) in the name.

Version

Version of the tool suite, for example V6R2016x or IC5141

Note: For DesignSync tool suites, the version must be of the form `V6Rnnnnn.*`, matching the released version of DesignSync. If you are not sure what the version string is, type "version," from the command line of any DesignSync client.

Type

The type of tool suite.

- **DesignSync** - tool suites containing DesignSync tools. A development cannot be enabled successfully unless there is at least one DesignSync-type tool suite associated for the development. Every assignment created for a development must have defined at least one DesignSync-type tool suite that it can use. The user specifies an assignment when making a development area. The assignment's DesignSync-type tool suite dictates which version of DesignSync to use to initially populate the user's development area.
- **Custom type** - enter a string which can be used by any custom scripts to identify the tool suites by type.

Platform

Select the desired platform from the pull-down list of platforms that DesignSync supports. To create a tool suite for multiple compatible platforms select All, UNIX or Windows, as desired.

Path

Enter the path name to the tools on the platform. This is required when the tool is of the [Type](#) DesignSync. You can use environment variables in your path.

For DesignSync type tool suites, this value is used to set SYNC_DIR for the development areas that use this tool suite. DesignSync sets the shell variable `sync_toolsuite_path` to this value prior to sourcing the [Script](#) for other tool suites.

Notes:

- The path setting for DesignSync should be the *installation directory* rather than the *SYNC_DIR*. For pre-V6R2013 releases this is the same thing, but for V6R2013 and subsequent releases, the SYNC_DIR includes the platform name at the end. The EDA system automatically adds the platform directory to the path if necessary.
- The shell variable, `sync_toolsuite_path`, is not an environment variable and is not visible with the commands that allow you to examine the environment variables.

Script

Path to scripts that run before the tool is started to set the environment for this tool suite. Value may include environment variables. For UNIX, you must use a bourne shell script (.sh). For Windows, this must be a DOS batch file (.BAT).

When this script is sourced, the tool suite [Path](#) is available in the shell variable `sync_toolsuite_path` (this allows scripts to be used for different installation/versions of the tool suite.) This script is sourced (not run), so no parameters are passed in. The script cannot accept user input at run-time. It must be able to be processed without user input.

Note: For a DesignSync tool suite type, if a specific SYNC_CUSTOM_DIR is needed, use this script to set the value.

The following environment or shell variables are set or available to unset within the script:

- SYNC_DEVELOPMENT_DIR - top of the development instance directory
- SYNC_DEVAREA_DIR - development areas directory

- `SYNC_DEVAREA_TOP` - the leaf name of the top module or directory in the development area
- `SYNC_PROJECT_CFGDIR` – top of the development instance's setting directory
- `SYNC_PROJECT_CFGDIR_ROOT` – root directory for the setting module
- `SYNC_DEV_ASSIGNMENT`– name of assignment for the development area. Note that this can be ""
- `sync_toolsuite_path` – path to the tool. Note that this is not an environment variable and is not visible with the commands that allow you to examine the environment variables.
- `SYNC_DIR` – DesignSync tools installation directory. This is for the DesignSync tool suite for this development

Note: By default, tools in the tool suite are not searched for in the tool suite [Path](#). Tools should use full absolute paths for their [Program name](#) or add the tool suite path to their environment in the tool suite script, for example:

```
PATH=$sync_toolsuite_path:$PATH
export PATH
```

Caution: The scripts for tool suites are not run in a a specific order. If you have executables with the same name in multiple tool suite paths and you set your PATH in this manner for each tool suite, you can not predict which executable will be run.

Program Group

The Program Group for this tool suite. This must be a valid windows Program Group value. This may be a relative path. This is required for tool suites running on Windows.

Starting directory

Defines a path relative to the development area directory which the system will “cd” to before running the tools. (This value can be overridden by the individual tool setting.) This may be a relative path and/or contain environment variables.

Description

Enter a text description of a tool suite.

Return

Save the changes in memory and return to the previous page.(Tool Suites only)

Submit

Save the changes to the database and return to the previous page. (Shared Tool Suites only)

Cancel

Cancel the changes and return to the previous page.

Help

Launches a web-browser and opens this help page.

Edit Tool Suite/View or Edit Shared Tool Suite

This page contains the Edit Tool Suite page, the View or Edit Shared Tool Suite page, and the sub-object table for tool definitions.

DesignSync Developments | Edit Tool Suite of Development "

Complete this form and press 'Return' to record the changes and return to the Add Development page, or press 'Cancel' to return without making any changes.

<p>Name*</p> <p>Version*</p> <p>Type</p> <p>Platform</p> <p>Path</p> <p>Script</p> <p>Program group</p> <p>Starting directory</p> <p>Description</p>	<p>* Denotes a required field</p> <p>EditingTool</p> <p>x5</p> <p> <input type="radio"/> DesignSync <input checked="" type="radio"/> Custom type: <input style="width: 150px;" type="text" value="Documentation Editor"/> </p> <p>All</p> <p><input style="width: 150px;" type="text" value="/usr/contrib/bin/doceditor"/></p> <p><input style="width: 150px;" type="text" value=""/></p> <p><input style="width: 150px;" type="text" value="Preferred Documentation Editor"/></p> <p><input style="width: 150px;" type="text" value=""/></p> <p><input style="width: 150px; height: 40px;" type="text" value=""/></p>
---	--

Tools

Display String	Edit	Description	Program name	Arguments	Graphical	Starting directory
<input type="checkbox"/>	DesignSync	DesignSync GUI tool	DesignSync GUI (x64)		✓	
<input type="checkbox"/>	DocEditor	Tool used by the technical writers to create documentation	Documentation Editing Tool		✓	

Click on the image for more information about each field.

Notes

- When working on the Shared Tool Suites, the Submit button replaces the Return button depicted above. The Submit button saves the tool suite definition in the database making it available to all developments using the Shared Tool Suites. Any developments currently using the tool suite may need to be restarted to use the new version.
- The Return button available for Tool Suites that are not shared, updates the Tool Suite by modifying the in-memory definition.
- The View Shared Tool Suites page includes an Add button which launches the [Add Shared Tool Suite](#) page.

Field Descriptions

This section describes the field available on this panel.

Name

Name of the tool suite. This field is not editable. If the tool suite name must be changed, then a new tool suite with the new name must be added and the old tool suite deleted.

Note: The remaining fields of the Edit Tool Suite panel essentially behave the same as the Add Tool Suite panel.

Version

Version of the tool suite, for example V6R2014x or IC5141

Note: For DesignSync tool suites, the version must be of the form V6Rnnnnn.*, matching the released version of DesignSync. If you are not sure what the version string is, type "version," from the command line of any DesignSync client.

Type

The type of tool suite.

- **DesignSync** - tool suites containing DesignSync tools. A development cannot be enabled successfully unless there is at least one DesignSync-type tool suite associated for the development. Every assignment created for a development must have defined at least one DesignSync-type tool suite that it can use. The user specifies an assignment when making a development area. The assignment's DesignSync-type tool suite dictates which version of DesignSync to use to initially populate the user's development area.
- **Custom type** - enter a string which can be used by any custom scripts to identify the tool suites by type.

Platform

Select the desired platform from the pull-down list of platforms that DesignSync supports. To create a tool suite for multiple compatible platforms select All, UNIX or Windows, as desired.

Path

Enter the path name to the tools on the platform. This is required when the tool is of the [Type](#) DesignSync. You can use environment variables in your path.

For DesignSync type tool suites, this value is used to set SYNC_DIR for the development areas that use this tool suite. DesignSync sets the shell variable `sync_toolsuite_path` to this value prior to sourcing the [Script](#) for other tool suites.

Notes:

- The path setting for DesignSync should be the *installation directory* rather than the *SYNC_DIR*. For pre-V6R2013 releases this is the same thing, but for V6R2013 and subsequent releases, the SYNC_DIR includes the platform name at the end. The EDA system automatically adds the platform directory to the path if necessary.
- The shell variable, `sync_toolsuite_path`, is not an environment variable and is not visible with the commands that allow you to examine the environment variables.

Script

Path to scripts that run before the tool is started to set the environment for this tool suite. Value may include environment variables. For UNIX, you must use a bourne shell script (.sh). For Windows, this must be a DOS batch file (.BAT).

When this script is sourced, the tool suite [Path](#) is available in the shell variable `sync_toolsuite_path` (this allows scripts to be used for different installation/versions of the tool suite.) This script is sourced (not run), so no parameters are passed in. The script cannot accept user input at run-time. It must be able to be processed without user input.

Note: For a DesignSync tool suite type, if a specific SYNC_CUSTOM_DIR is needed, use this script to set the value.

The following environment or shell variables are set or available to unset within the script:

- SYNC_DEVELOPMENT_DIR - top of the development instance directory

- SYNC_DEVAREA_DIR - development areas directory
- SYNC_DEVAREA_TOP - the leaf name of the top module or directory in the development area
- SYNC_PROJECT_CFGDIR – top of the development instance's setting directory
- SYNC_PROJECT_CFGDIR_ROOT – root directory for the setting module
- SYNC_DEV_ASSIGNMENT– name of assignment for the development area. Note that this can be ""
- sync_toolsuite_path – path to the tool. Note that this is not an environment variable and is not visible with the commands that allow you to examine the environment variables.
- SYNC_DIR – DesignSync tools installation directory. This is for the DesignSync tool suite for this development

Note: By default, tools in the tool suite are not searched for in the tool suite [Path](#). Tools should use full absolute paths for their [Program name](#) or add the tool suite path to their environment in the tool suite script, for example:

```
PATH=$sync_toolsuite_path:$PATH
export PATH
```

Caution: The scripts for tool suites are not run in a a specific order. If you have executables with the same name in multiple tool suite paths and you set your PATH in this manner for each tool suite, you can not predict which executable will be run.

Program Group

The Program Group for this tool suite. This must be a valid windows Program Group value. This may be a relative path. This is required for tool suites running on Windows.

Starting directory

Defines a path relative to the development area directory which the system will “cd” to before running the tools. (This value can be overridden by the individual tool setting.) This may be a relative path and/or contain environment variables.

Description

Enter a text description of a tool suite.

Return

Save the changes in memory and return to the previous page.(Tool Suites only)

Submit

Save the changes to the database and return to the previous page. (Shared Tool Suites only)

Cancel

Cancel the changes and return to the previous page.

Help

Launches a web-browser and opens this help page.

Related Topics

[Add Tool Suite/Add Shared Tool Suite](#)

[View Tool Suite](#)

View Tool Suite/View Shared Tool Suites

This page displays the View Tool Suite/View Shared Tool Suite page and the Tool/Shared Tool sub-objects associated with the tool suite. This page is accessible by clicking the name of the Tool Suite on the [View/Edit Development](#) page.

To view the properties associated with a sub-object, click the Name of the sub-object. .

To finish viewing the tool suite, press the **Ok** button.

To edit the tool suite, press the **Ok** to return to the [View/Edit Development](#) page and select the **Edit** icon.

Related Topics

[Edit Development](#)

[View/Edit Developments](#)

[View Development](#)

[View Development Areas](#)

[View Development Settings](#)

[View Tool](#)

[View Development Assignment](#)

Understanding Tools/Shared Tools

A tool is the definition of a single application to be run against the development's design data by an individual user. The tools are associated with a development by means of a tool suite. Shared Tools are available to Shared Tool Suites and can be used by any development with the correct permissions. Tools that are not shared are only available to the development in which their Tool Suite is defined.

Related Topics

[Edit Tool](#)

[Edit Tool Suite](#)

[View/Edit Developments](#)

Add Tool/Add Shared Tool

Tool Suites manage the Tools (programs and scripts) available to the users. There is no difference between the options on the Add Tool and the Add Shared Tool panels.

Note: You can define an OS shell as a tool, however, if you do, you will want to disable any startup scripts (.profile, .cshrc, etc) to prevent the script from altering the environment unpredictably. You can use the [Script](#) field in the Tool Suite definition to pass along any environment settings you need to set for commands within the shell to function properly.

Click on the image for more information about each field.

DesignSync Developments | Add Tool to Tool Suite 'EditingTool' of Development "

Complete this form and press 'Return' to record the changes and return to the Add Tool Suite to Development " page, or press 'Cancel' to return without making any changes.

	* Denotes a required field
Display String*	<input type="text" value="DocEditor"/>
Graphical	<input checked="" type="checkbox"/>
Program name*	<input type="text" value="Documentation Editing Tool"/>
Arguments	<input type="text"/>
Starting directory	<input type="text"/>
Description	<input type="text" value="Tool used by the technical writers to create documentation"/>

Field Descriptions

This section describes the field available on this panel.

Display String

Display name for the program being run. When the user sets their environment to run on a development area, they also select a tool to run on that area. The Display String is the name they will see when selecting to tool to invoke. The Display String must be unique within the tool suite, but is not required to be unique within the entire system. If your development includes tool suites that contain tools of the same name, the user will be asked to identify the tool using both the tool suite [Name](#) and the tool display string.

Graphical

Indicates whether this tool has a graphical interface or a command line interface. Tools with a graphical interface require a graphical windowing environment to invoke the tool. Non-graphical tools can be run from a simple text-based command line environment such as telnet. Click the checkbox to indicate a graphical tool. Unclick the checkbox to indicate a command-line tool.

Program name

Enter the command to run this tool.

For UNIX systems , enter the name of the executable.

For example, to run the DesignSync GUI client, you would enter: `DesSync`

For Windows systems, enter the name of the program in the program group.

For example, to run the DesignSync GUI client, you would enter: **DesignSync GUI (x64)**

Arguments

Enter the arguments to pass to the program. This is run in addition to the settings set by the startup [Script](#) defined for the Tool Suite.

Note: If you are defining an OS shell as a tool, you can use this field to disable any startup scripts; for example, if you are using csh, you can pass the -f argument to disable running the .cshrc.

Starting directory

The directory to set as the current working directory (cwd) for the tool when the tool starts. If a starting directory is defined here, it will override the [Starting directory](#) defined for the Tool Suite. The user connecting to the development may override this setting by specifying a different starting directory when they connect to their development area.

Description

Enter a text description of the tool.

Return

Save the changes and return to the previous page.

Cancel

Cancel the changes and return to the previous page.

Help

Launches a web-browser and opens this help page.

Related Topics

[Edit Tool](#)

[View Tool](#)

[Add Tool Suite](#)

[View/Edit Development](#)

Edit Tool/Edit Shared Tool

You can modify most of the options for a defined Tool or Shared tool using the Edit Tool or Edit Shared Tool panel. You open this panel by selecting the Edit icon for the tool being edited from the sub-object table on the [Edit \(Shared\) Tool Suite](#) page.

Click on the image for more information about each field.

DesignSync Developments | Edit Development Assignment of Development "

Complete this form and press 'Return' to record the changes and return to the Add Development page, or press 'Cancel' to return without making any changes.

Name* * Denotes a required field
DocWriter

Development Selector
 Default selector
 Custom selector:

Development Setting ChipSettings ▾

Development View Doc

Description

Tool Suites

Tool Suites
 Use all development's tool suites
 Check mark tool suites to use from the table

	Name	Type	Version	Description	Platform	Development path
<input type="checkbox"/>	EditingTool	Documentation Editor	X5		All	/usr/contrib/bin/doceditor

Field Descriptions

This section describes the field available on this panel.

Display String

Display name for the program being run. This field is not editable. If the display name must be changed, then a new tool must be added with the new name and the old Tool deleted.

Note: When the user using the development selects a tool from a list or provides a tool name, this is the tool name they will provide. The Display String must be unique within the tool suite, but is not required to be unique within the entire system. If your development includes tool suites that contain tools of the same name, the user will be asked to identify the tool using both the tool suite [Name](#) and the tool display string.

Note: The remaining fields of the Edit Tool panel essentially behave the same as the Add Tool panel.

Related Topics

[AddTool](#)

[View Tool](#)

[Edit Tool Suite](#)

[View/Edit Development](#)

View Tool

This page displays the View Tool page. This page is accessible by clicking the name of the Tool on the [View Tool Suite](#) page.

To finish viewing the tool, press the **Ok** button.

To edit the tool, press the **Ok** to return to the [View Tool Suite](#) page and select the **Edit** button.

Development Assignments

A development assignment defines a way of working on development data. Every development area created by an engineer is associated with a particular assignment.

The assignment defines what data is fetched into a development area and how this data is fetched and used. A limited set of tools may also be defined for use by an assignment. A set of assignments may be defined for a development to customize the development area for each type of work being done.

For example, a developer might want access to the data and tools that apply only to the code developer, but a documentation specialist will want different tools and a different view of the documentation files. The development can be defined to serve both of those users, and the Development assignment provides customization that applies to the development areas of that development.

Another example might be a developer who wants to maintain two separate views, a static view of the data, for reference, and a dynamic view for work in progress. In a SITaR style environment, that developer might be performing both designer tasks, requiring a work-in-progress view, and integrator tasks requiring a static view.

Each assignment may have associated with it a setting, a DesignSync selector, a view and a set of tool suites.

If no assignment is created, a "Generic" assignment is created and used as the default.

Add Development Assignment

This page contains the Add Development Assignment page.

Click on the image for more information about each field.

DesignSync Developments | Add Development Assignment to Development "

Complete this form and press 'Return' to record the changes and return to the Add Development page, or press 'Cancel' to return without making any changes.

Name* * Denotes a required field

DocWriter

Development Selector

Default selector

Custom selector:

Development Setting ChipSettings ▾

Development View Doc

Description

Tool Suites

Tool Suites

Use all development's tool suites

Check mark tool suites to use from the table

	Name	Type	Version	Description	Platform	Development path
<input type="checkbox"/>	EditingTool	Documentation Editor	X5		All	/usr/contrib/bin/doceditor

Field Descriptions

This section describes the fields available on this panel.

Name

Name of the development assignment. This name must be unique within the development.

Note: For optimal function, you should avoid using any [reserved characters](#) in the name.

Development Selector

Determines whether to use the selector defined with the development to populate the design data (the default) or a different selector.

- Select Default Selector to use the development's selector.
- Select Custom Selector and enter the desired selector value to use a different selector.

You should distinguish between branch and version selectors. If you are specifying a branch other than Trunk, specify both the branch and version as follows:

'<branchtag>:<versiontag>', for example,

'Rel2:Latest'

Development Setting

Determines whether to use the setting defined with the development or specify a different setting. The available development settings are listed in a drop-down box.

Select the desired development setting, or select <Default> to use the defined development setting default. The default is shown on the [View/Edit Developments](#) page.

Development View

If your development's design data is modules-based, you can optionally enter a modules view to restrict the data populated for the development areas using this assignment. For more information using module views, see the *ENOVIA Synchronicity DesignSync Data Manager User's Guide: [Understanding Module Views](#)*.

Description

Description of the development assignment.

Tool Suites

Select the Tool Suites that apply to the development assignment. You can select from the following options:

- Use all development's tool suites - Automatically uses all the tool suites defined for the development.
- Check mark tool suites to use from the table - Select the desired tool suites from the Tools Suite table below the heading.

View Tool Suite Properties

The Tool Suite table provides a quick view of the properties of the tool suites defined for the development. Clicking on the Tool Suite name opens the [View Tool Suite](#) page..

Return

Save the changes and return to the previous page.

Cancel

Cancel the changes and return to the previous page.

Help

Launches a web-browser and opens this help page.

Related Topics

[Development Assignments](#)

[Edit Development Assignment](#)

[View Development Assignment](#)

[View/Edit Developments](#)

[Add Developments](#)

[Add Developments Settings](#)

[Add Tool Suite](#)

[Add Tool](#)

Edit Development Assignment

This page contains the Edit Development Assignment form.

Click on the image for more information about each field.

DesignSync Developments | Edit Development Assignment of Development "

Complete this form and press 'Return' to record the changes and return to the Add Development page, or press 'Cancel' to return without making any changes.

Name* * Denotes a required field
DocWriter

Development Selector
 Default selector
 Custom selector:

Development Setting ChipSettings ▾

Development View Doc

Description

Tool Suites

Tool Suites

Use all development's tool suites
 Check mark tool suites to use from the table

	Name	Type	Version	Description	Platform	Development path
<input type="checkbox"/>	EditingTool	Documentation Editor	X5		All	/usr/contrib/bin/doceditor

Field Descriptions

This section describes the field available on this panel.

Name

Name of the development assignment. This field is not editable. If the assignment name needs to be changed, then a new assignment should be added and the old assignment deleted.

Note: The remaining fields of the Edit Development Assignment essentially behave the same as the [Add Development Assignment](#) panel.

Related Topics

[Development Assignments](#)

[Add Development Assignment](#)

[View Development Assignment](#)

[View/Edit Developments](#)

[Edit Development](#)

[Edit Development Settings](#)

[Edit Tool Suite](#)

[Edit Tool](#)

View Development Assignment

This page displays the View Development Assignment page and the associated tool suite information. This page is accessible by clicking the name of the development assignment on the [View/Edit Development](#) page.

To view the properties associated with a sub-object, click the Name of the sub-object. .

Note: If [Use all development's tool suites](#) is selected for the development assignment, you do not see a list of the tool suites in the Tool Suites sub-object table.

To finish viewing the development assignment, press the **Ok** button.

To edit the development assignment, press the **Ok** to return to the [View/Edit Development](#) page and select the **Edit** button.

Related Topics

[View/Edit Development](#)

[Add Development Assignment](#)

[Edit Development Assignment](#)

Managing Development Areas

Setting up Development Areas

In order to take advantage of the capabilities of the Enterprise Design Administration functionality, a user creates a local development area or joins a shared development area and launches the tools available through the defined development assignment to work with the data contained in the development.

The development area is configured, maintained, and used through the DesignSync Development Areas (sda) interface. This interface is available in both graphical and command line forms. The command-line version contains its own in-line help. For more information on using the command line interface for the Development Areas, see the [ENOVIA Synchronicity Command Reference](#).

Prerequisites

Before a user can use a development area, the user must have:

- one or more development servers listed in the DesignSync Administration tool. For information on defining a development server, see the *ENOVIA Synchronicity DesignSync Administrator's Guide*: [Development Servers](#).
- one or more developments defined with Enterprise DesignSync Administration. For more information on defining developments, see [Add Development](#).

Getting Started Using Development Areas

To get started using Development Areas, you need to create or attach to a Development Area using the sda tool.

Starting the Development Area Management Tool from a Shell:

To invoke the Development Area GUI to work with EDA development areas, enter this command at the operating system's shell prompt:

```
sda gui
```

Note: This command is not valid within a DesignSync client shell. If tried, an error notifies the user that *sda* must be invoked from the OS shell.

In addition, the other *sda* commands (*mk*, *ls*, *rm*) have a `-gui` option that when used, invokes the *sda* GUI starting on the appropriate tab related to the *sda* command being invoked. If there are other options given to the command, then the value of those options are also carried forward into the fields of the GUI tab. For more information on the *sda* commands, see [ENOVIA Synchronicity Command Reference: sda](#).

Starting the Development Area Management Tool from the Windows Start Menu:

On Windows, open the Program Group that points your DesignSync client installation and select **DesignSync Development Areas (x64)**.

Related Topics

[List Areas](#)

[Make Area](#)

[Remove Area](#)

[Join Area](#)

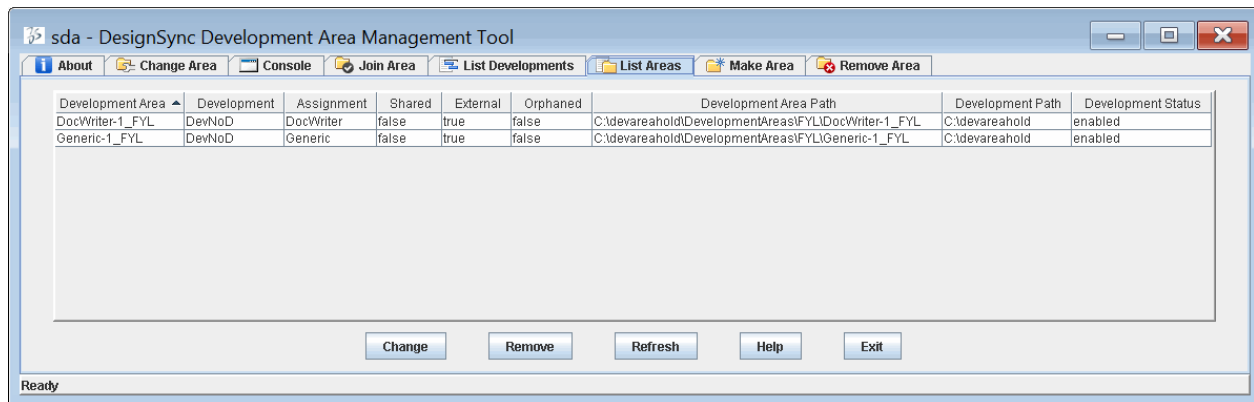
[List Developments](#)

[Change Area](#)

List Areas

List Areas lists the areas that are currently active for the user.

The list displays information about the development area. Each column in the table is sortable. Click the column header to sort alphabetically, or double-click to sort reverse-alphabetically. An arrow will appear in the sorted column showing the direction of the sort order.



Explanation of Fields

Development Area

Name of the development area.

Development

Name of the development.

Assignment

Name of the development assignment.

Shared

Indicates whether the area is designated as shared. It does not indicate whether more than one user is using it.

External

Indicates whether the development area is external.

Orphaned

Indicates if the development area is orphaned. A development area is considered to be orphaned if the development area does not exist on disk, if the development definition does not exist on disk, or if the development area is not listed in its corresponding development definition.

Development Area Path

Local path to the development area.

Development Path

Path to the local directory containing the development definition

Development Status

Indicates the status of the development. The development may be enabled, disabled, or deleted.

Change

When a development area is selected, you can click the Change button to switch to the Change Area tab with the selected development area active.

Remove

When one or more development areas are selected, the Remove button allows you to delete the development area. When a single development area is selected for removal, DesignSync will verify that the area does not contain locked or modified files. If there are modified or locked files, the user is prompted to force removal, or cancel the operation.

Important: If multiple development areas are selected, DesignSync will not do any checks on the developments to determine if the areas contains locked or modified files. The user will not be prompted to complete the removal.

Refresh

Refreshes the Development Area settings. The user is required to confirm the operation.

Help

Opens this help page in your web browser.

Exit

Closes the DesignSync Development Area Management tool..

Related Topics

[List Developments](#)

Make Area

Make area creates a new development area in the specified location and registers the development area with the development server managing the development. The development server and the development the area uses must already exist. For more information on defining a development server, see the *ENOVIA Synchronicity DesignSync Administrator's Guide: [Development Servers](#)*. For more information on creating a development , see [Add Development](#).

Make area performs the following sequence of actions:

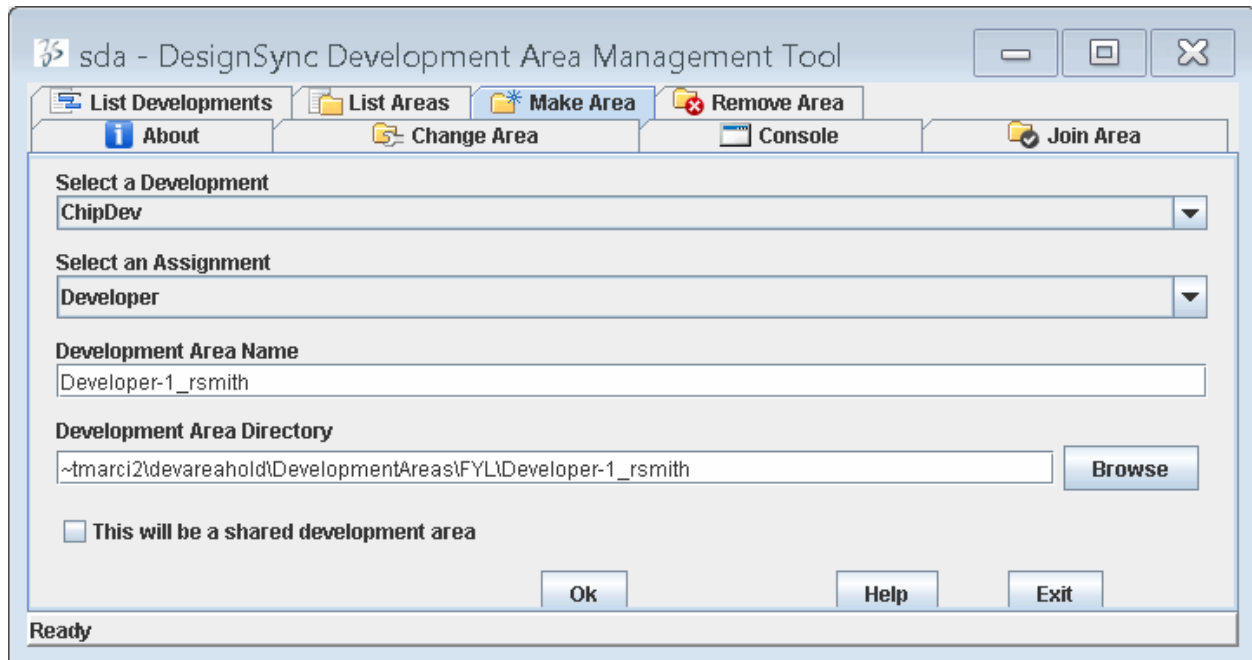
1. Creates the development area directory, if necessary.

2. Sets up the environment by creating environment variables to point to the new development area. The environment variables are:
 - SYNC_DEVAREA_DIR - the new development area directory.
 - SYNC_DEVELOPMENT_DIR - the top-level of the development instance directory.
 - SYNC_PROJECT_CFGDIR - the setting for the assignment associated with the development area.
3. Runs any defined PreCommand triggers.
4. Populates the development area with the development's data using the development URL from the development instance definition; the selector from the assignment associated with the development area; the version of DesignSync tools specified with the assignment; and any settings specified in the setting for the assignment, for example, the fetch state setting. The development data is populated into a sub-directory of the development area named using the leaf name of the containing server data.
5. Runs any defined PostCommand triggers.

Notes:

- For Windows development areas, the fetch state is automatically set to -get mode (Fetch Unlocked Copies).
- Server access may require a username and password. If your password for the server is not already saved by the client, you may be prompted to enter it in order to access the server data. For more information, see [Accessing a SyncServer with User Authentication](#).
- After the development area has been created, the tool automatically switches to the [Change Area](#) tab so you can use the newly created development area.

Click on the image for more information about each field.



Field Descriptions

This section describes the field available on this panel.

Select a Development

Select the development from the list of valid available developments. If you expect to see a name in this list, but do not see it, check to make sure the development has been enabled.

Select an Assignment

Select the development assignment from the list of valid development assignments. If you expect to see a name in this list, but do not see it, verify that the development assignment is in the development you selected.

Development Area Name

Enter a name that uniquely identifies this development area.

A development area name must start with an alphanumeric character and be composed of alphanumeric characters, including dot (.), dash (-), or underscore (_). Development names must be unique within a development server. Development area names must be unique for a development instance.

By default, the development area name is prefilled with [Assignment](#) name, the "count", and the user name of the user creating the development. The count begins at one and increments for each development containing the same assignment created by the same user, thus allowing the development area name to remain unique.

Notes:

- Development areas are checked for uniqueness in the name/instance pair. You cannot have two development areas for the same instance using the same name. You can have two development areas with the same name if they are for different development instances.
- You cannot use any [reserved characters](#) in the name.

Development Area Directory

Local path to the development area. This is automatically prefilled based on the defined [Development Area root](#).

This will be a shared development area

Indicates whether the area is intended to be shared among multiple users. Click the checkbox if the area is intended to be shared. Unclick the checkbox if the development area is intended for a single user only (Default).

Note: If your administrator has disabled shared development areas, you will not see this option.

Make Development Area

Click this button to create the development area.

Ok

Apply selections.

Help

Opens this help page in your web browser.

Exit

Closes the DesignSync Development Area Management tool.

Related Topics

ENOVIA Synchronicity Command Reference: [sda mk](#)

[Join Area](#)

[Remove Area](#)

DesignSync Data Manager Administrator's Guide: [Make Area \(sdamk\) Command Triggers](#)

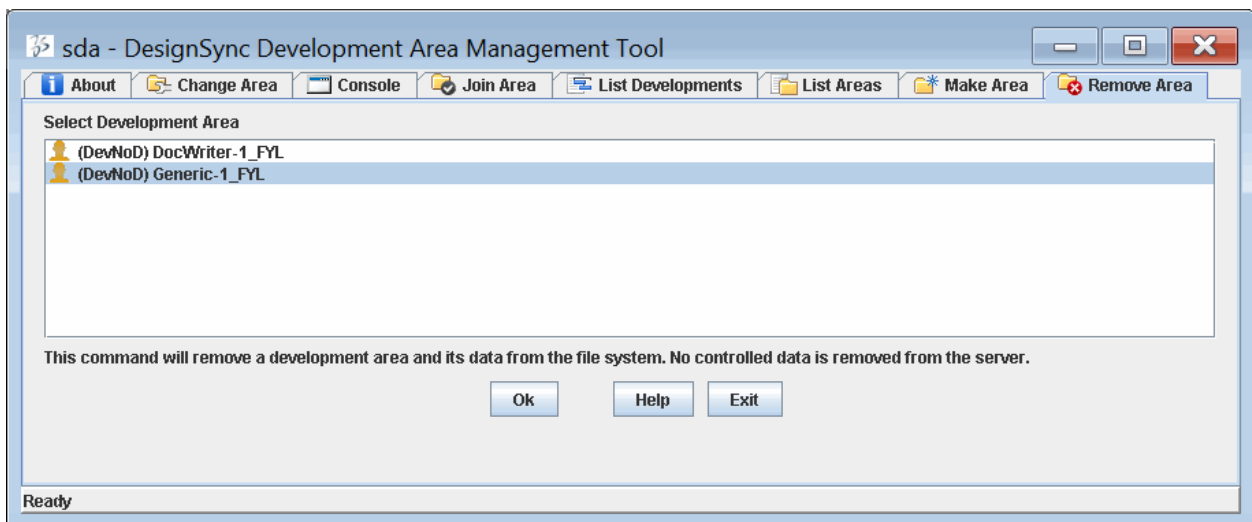
Remove Area

Remove Area removes a development area from the client and attempts to remove the local development area directory. If the development area is a shared development area, the local development area directory is not removed as long as there are still local areas using it. The operation does not remove any design data from the repository server.

The Remove Area operation uses the DesignSync rmfolder command to remove the development area folder. DesignSync provides the ability to run a command trigger for the remove area operation which is keyed to the rmfolder operation. PreCommand triggers run before the rmfolder operation. PostCommand triggers run after the rmfolder operation. For more information, see *DesignSync Data Manager Administrator's Guide: [Remove Area \(sdarm\) Command Triggers](#)*.

Note: Using Remove Area, you can remove a single area at a time, with all the checks associated with removing a development area. Using the [List Areas](#) tab, you can remove multiple areas, although it does not perform the usual checks.

Click on the image for more information about each field.



Field Descriptions

This section describes the field available on this panel.

Select Development Area

Select the development area from the list of known development areas.

Remove Area

Click this button to remove the development area.

Help

Opens this help page in your web browser.

Exit

Closes the DesignSync Development Area Management tool.

Related Topics

[Make Area](#)

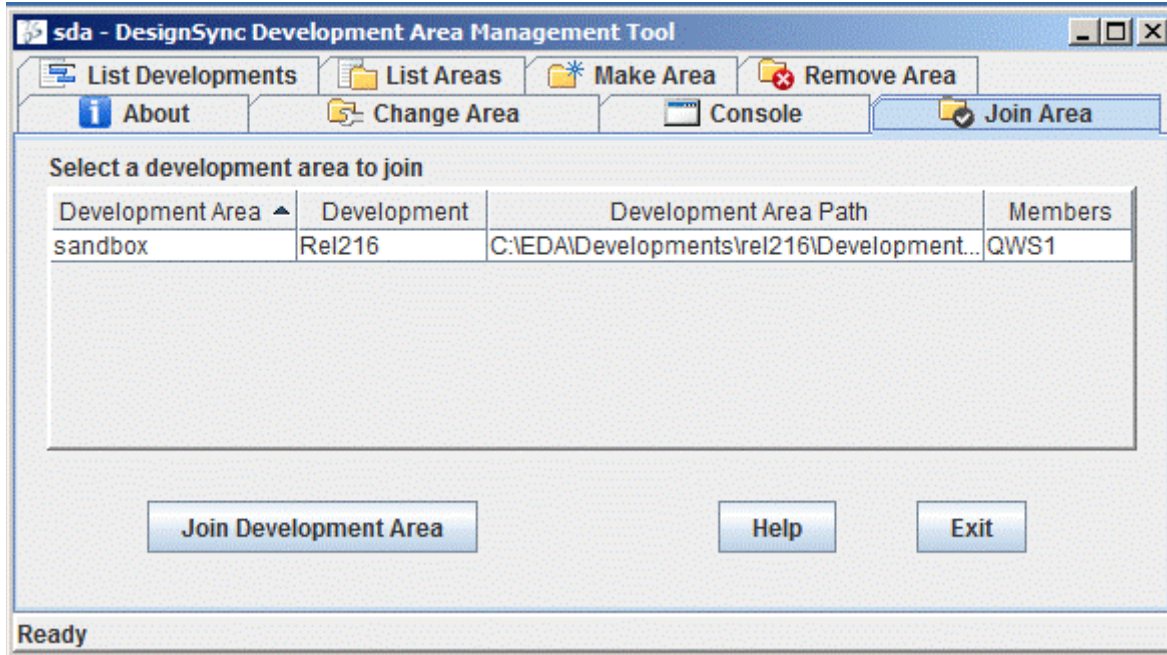
[List Areas](#)

DesignSync Data Manager Administrator's Guide: [Remove Area \(sdarm\) Command Triggers](#)

Join Area

Join Area allows the user to join an existing eligible shared area of a development. Eligible shared areas are located by finding the participating development servers from the registry, looking at the developments on those servers and identifying the shared areas that have a local path and have not already been joined. For information on defining a development server, see the *ENOVIA Synchronicity DesignSync Administrator's Guide*: [Development Servers](#).

Click on the image for more information about each field.



Field Descriptions

This section describes the field available on this panel.

Select a development area to join

Select the development area from the list of valid shared development areas.

Join Development Area

Click this button to join the development area.

Help

Opens this help page in your web browser.

Exit

Closes the DesignSync Development Area Management tool.

Related Topics

ENOVIA Synchronicity Command Reference: [sda_join](#)

[Make Area](#)

List Developments

List Developments lists developments that are currently active for the user, or registered with the development servers defined in SyncAdmin. For more information on defining development servers, see the *ENOVIA Synchronicity DesignSync Administrator's Guide*: [Development Servers](#).

The list displays information about the developments. Each column in the table is sortable. Click the column header to sort alphabetically, or double-click to sort reverse-alphabetically. An arrow will appear in the sorted column showing the direction of the sort order.

Development	Assignments	Data URL	Selector	Development Path	Server URL	Status
ChipDev	Developer	sync://wrrh17mon:30126/Modules/ChipDev/ChipNZ3	Trunk:	D:\devareahold	sync://wrrh17mon.dsone.3ds.com:30126	enabled
DevNoD	Generic	sync://wrrh17mon.dsone.3ds.com:30126/Modules/ChipNZ3	Trunk:	C:\devareahold	sync://wrrh17mon.dsone.3ds.com:30126	enabled

Explanation of Fields

Development

Name of the development.

Assignments

List of all assignments defined with the development.

Data URL

Location of the data on the server. This will be the module for a modules-based development, or containing folder for a files-based development.

Selector

Default selector of the data in the development.

Development Path

Path to the local directory containing the development definition.

Server URL

The hostname(s) and port number(s) of the development server(s).

Status

Indicates whether the status of the development. The development may be enabled or disabled.

Make Area

If a development is specified, switches to the [Make Area](#) tab with that development selected as the source development for the new area.

Help

Opens this help page in your web browser.

Exit

Closes the DesignSync Development Area Management tool..

Related Topics

[List Areas](#)

Change Area

Change Area allows the user to launch a tool from a development area which has already been set up on your system. The tool runs using the development setting defined for the area.

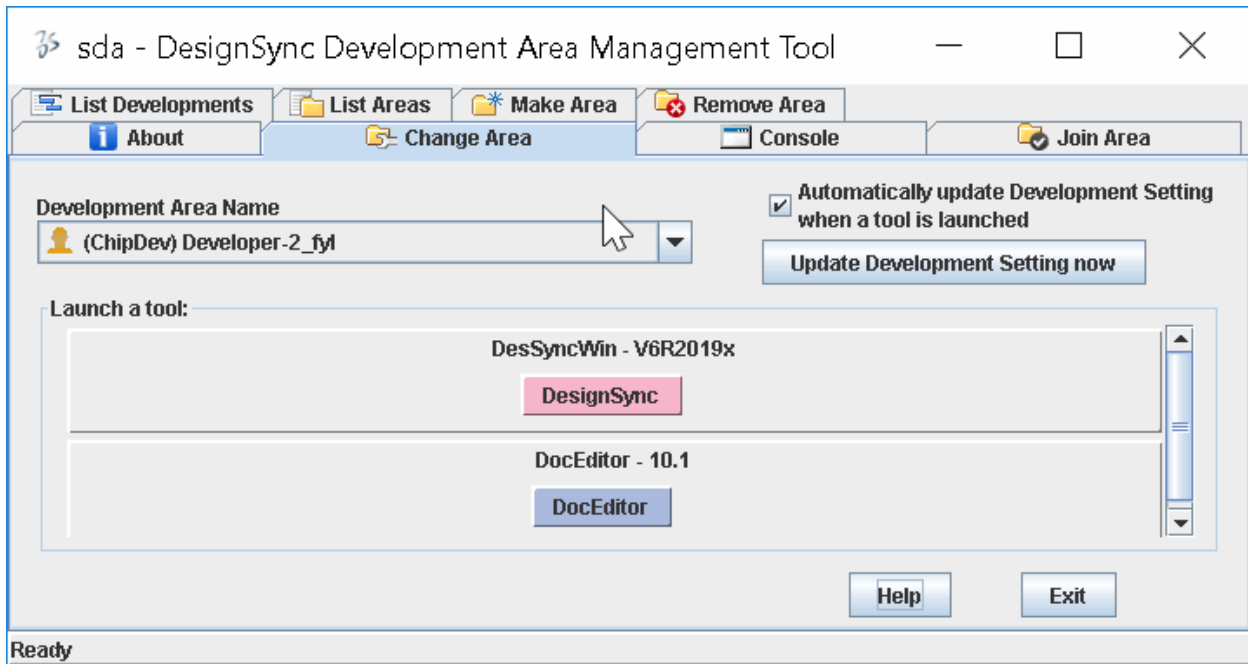
The Change Area utility performs the following sequence of actions:

1. If the **Automatically update Development Setting when a tool is launched** option is selected, updates the development instance directory associated with an external development area.
2. Sets up the environment by setting the following environment variables:
 - SYNC_DEVAREA_DIR - set to the requested development area directory.

- SYNC_DEVAREA_TOP - set to the leaf name of the top module or directory in the development area.
 - SYNC_DEV_ASSIGNMENT - set to the assignment associated with the development area
 - SYNC_DEVELOPMENT_DIR - set to the top of the development instance directory.
 - SYNC_PROJECT_CFGDIR - set to the directory holding the development setting for the assignment associated with the development area.
 - SYNC_WS_DEVAREA_TOP - set to the leaf name of the top module or directory in the development area. This variable can then be used for the starting directory in any commands you construct within the specified tool.
 - Any additional environment variables defined in the development setting script.
3. Runs all of the set up scripts defined for the tools associated with the assignment. Running all the scripts is required to support inter-tool dependencies and shell tools.
Note: When a shell is defined as a tool, it should be defined to ignore the startup script for the shell. Any aliases, etc. defined in the shell startup script will not be available.
 4. Sets the current directory for the tool to the starting directory. The starting directory is the directory defined in the tool's definition. If no starting directory is specified, then the directory defined in the tool suite is used. If no starting directory is specified in the tool suite either, the development area is used.

The starting directories can be specified with environment variables and may be relative to the development area.
 5. Starts the requested tool.

Click on the image for more information about each field.



Field Descriptions

This section describes the field available on this panel.

Development Area Name

Select the desired development area from the list of available development areas. If you do not see the area you expect, verify that the area is not orphaned or that you have joined it. This option remembers the last selected value and opens with that value selected whether you are relaunching the tool or changing tabs.

Automatically update Development Setting when a tool is launched

Determines whether the development area connected to an external development is current and, if not current, updates the development area before launching the tool.

To verify that the external development is current, check this option.

To run the tool without verifying whether the external development is current, uncheck this option.

This option remembers the last selected value and opens with that value selected whether you are relaunching the tool or changing tabs.

Update Development Setting now

Click this button to verify that your development area is synchronized with your external development server and, if it is not current, update it.

Launch a tool

Lists the available tools as buttons. Press the button that corresponds to the tool you want to launch. The tool launches in the development setting pointing to the files in the development.

Help

Opens this help page in your web browser.

Exit

Closes the DesignSync Development Area Management tool.

Related Topics

ENOVIA Synchronicity Command Reference: [sda_cd](#)

Console

The Console window displays a portion of the information that is written to the log file specified at the top of the Console Window.

You cannot directly save output from the console window, but you can copy the text into a buffer on your Windowing system to paste into a file, should you need to save the output.

For more information on the operations run by the DesignSync Development Area Management Tool, open the log file specified at the top of the Console window.

Accessing a SyncServer with User Authentication

This topic describes the SyncServer dialog box, which appears when you connect to a development on a server on which user authentication has been enabled.

Your project leader may have configured DesignSync to require username and password authentication when accessing data on a SyncServer. The first time you access the server, DesignSync displays a dialog box where you must supply your username and password.

Username

Enter the username assigned to you by your system administrator. If you do not have a username, contact your system administrator to obtain one.

Note: Usernames may be case sensitive, depending on the system.

Password

Enter your password. If you do not have a password or have forgotten it, contact your system administrator.

Note: Passwords may be case sensitive, depending on the system.

Save password

Check this box to save your password locally. If you save your password, you will not be prompted for username and password again unless your password changes or expires.

If you do not save your password, you are prompted for the password every time you open a new client and connect to the specified server.

TroubleShooting

No Default User Specified

Problem: When activating a development, DesignSync issues the following error message:

```
ERROR: No default user specified  
createDevelopmentInstance failed with the message ERROR: No  
default user specified
```

Possible cause:The mirror system has not been set up with a default user.

Solution: Configure the mirror system. For information on configuring the data replication system, see [Setting Up Data Replication](#). For information on defining the default mirror user, see [Setting Up A Mirror Server](#).

Index

D

Data Replication Root 1, 5

Development 1

adding 5

data 18

editing 11, 14

instance 1

listing 55

server 1

settings 18

viewing 11, 14

Development Area 1

changing 56

getting started 45

joining 53

listing 46

making 48

path 46

removing 52

setting up 45

viewing 15

Development Assignments 37

adding 38

description 1

editing 41

viewing 43

Development Settings 17

adding 18

description 1

editing 21

viewing 23

E

Enterprise Design Administration

synchronization 3

Enterprise Design Synchronization

href 3

policy 3

tag 3

type 3

Enterprise DesignSync Administration 1

managing developments 5

T

Tool 33

adding 33

editing 36

viewing 37

Tool Suites 23

adding 24

editing 28

shared tool suites 16

use with tools 33

viewing 32

U

User Authentication 59