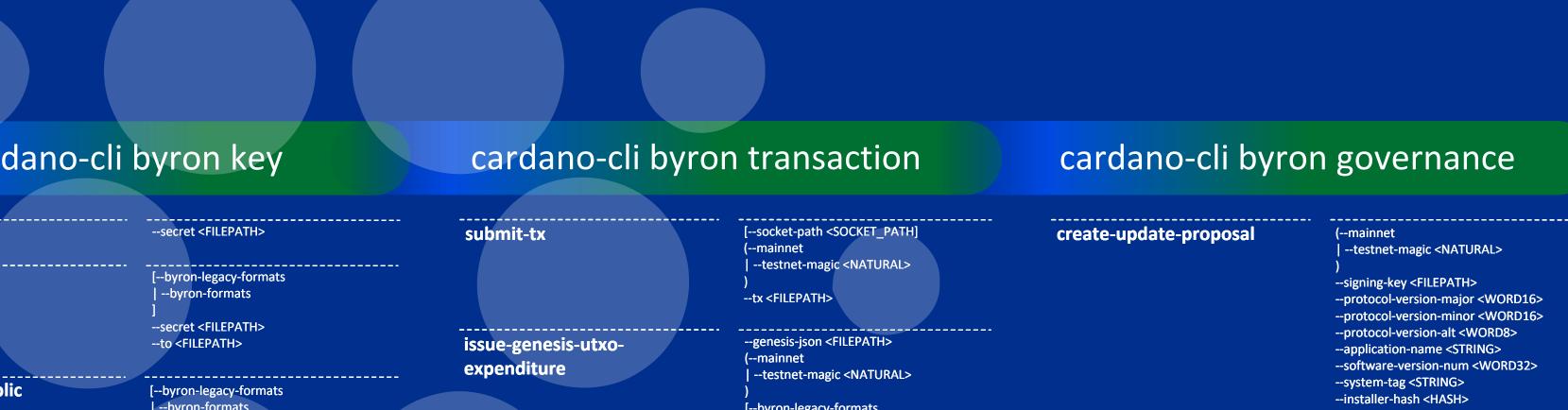
Cardano Command Line Interface V8.22.0.0









		expenditure	(mainnet testnet-magic <natural></natural>		software-versio
signing-key-public	[byron-legacy-formats byron-formats secret <filepath></filepath>) [byron-legacy-formats byron-formats] tx <filepath></filepath>		system-tag <stf installer-hash <i filepath <filepa [script-version < [slot-duration < [max-block-size</filepa </i </stf
signing-key-address	[byron-legacy-formats byron-formats] (mainnet testnet-magic <natural></natural>		wallet-key <filepath>rich-addr-from <addr> (txout '(<"ADDR", LOVELACE>)')</addr></filepath>		[max-header-siz [max-tx-size <n i="">[max-proposal-s [max-mpc-thd < [heavy-del-thd <</n>
)secret <filepath></filepath>	issue-utxo-expenditure	(mainnet testnet-magic <natural>) [byron-legacy-formats</natural>		[update-vote-th [update-propos [time-to-live <w [softfork-init-the</w
migrate-delegate-key-from	from <filepath>to <filepath></filepath></filepath>		byron-formats tx <filepath> wallet-key <filepath> (txin (<txid,index>)) (txout '(<"ADDR", LOVELACE>)')</txid,index></filepath></filepath>		softfork-min-t softfork-thd-d [tx-fee-a-consta tx-fee-b-const [unlock-stake-e
		txid	tx <filepath></filepath>	create-proposal-vote	(mainnet testnet-magic) signing-key <fil proposal-filepat</fil
cardano-cli b	yron genesis	cardano-cli byr	on miscellaneous		(vote-yes vote-no) output-filepath

	protocol-parameters-file <filepath>k <int>protocol-magic <int>n-poor-addresses <int>n-delegate-addresses <int>total-balance <int></int></int></int></int></int></filepath>		byron-tx byron-update-proposal byron-vote] filepath <filepath></filepath>	submit-proposal-vote	testnet-magic <natui) filepath <filepath> [socket-path <socket_i (mainnet</socket_i </filepath></natui
	delegate-share <double>avvm-entry-count <int>avvm-entry-balance <int> [avvm-balance-factor <double>] [secret-seed <int>]</int></double></int></int></double>	pretty-print-cbor	filepath <filepath></filepath>		testnet-magic <natur) filepath <filepath></filepath></natur
print-genesis-hash	genesis-json <filepath></filepath>				

) = Parenthesis. They represen] = Square brackets. They repr	the type of information to provide.					
	esent non-mandatory options.	/ TO BI		COMMAND		
= A separator that indicates a between several options in	ide brackets or parentheses.			COMMAND		
	First locate the command you will need. For our example we will take the Drep registration certificate from the		#1	cardano-cli	conway	governance drep
	conway top fonction. On the first line of your terminal you will therefore enter: cardano-cli conway governance drep registration certificate \		#2	→ registration-certific	cate	drep-verification-key <string> drep-verification-key-file <file></file></string>
					l	drep-key-hash <hash> key-reg-deposit-amt <natural> drep-metadata-url <text> drep-metadata-hash <hash> out-file <file></file></hash></text></natural></hash>
	Then on the next line of your terminal, you can start entering the command options. The first is a choice between 3 required options due to parentheses. For our example, we will use the DRep verification key file		Carda	no-cli conwa	ly govern	lance drep
	"drep.vkey"		registra	ation-certificate		on-key <string> ion-key-file <file> 1 <hash></hash></file></string>
	cardano-cli conway governance drep registration certificate \drep verification-key-file drep.vkey \	#3			[drep-metadat	ta-hash <hash>]</hash>
		• • • • • • •	• • • • • • • •		• • • • • • • • •	
	When there is no parentheses attached to an option and it		carda	ano-cli conwa	av gover	nance dren
	is part of the same column, the option is mandatory. So you need to use the next option which requires a deposit amount for your DRep registration certificate. The value of					Traffice di Cp
	the deposit must be in Lovelace.					tion-key <string></string>
			registi	ration-certificate	drep-verifica drep-key-ha	tion-key-file <file> sh <hash></hash></file>

you choose to use this option, you will be required to

provide the metadata hash of the sub-option located in the

The last option simply determines the name you want to give to your DRep registration certificate. "drep.cert"

cardano-cli conway governance drep registration certificate \
--drep verification-key-file drep.vkey \
--key-reg-deposit-amt 2000000 \

--drep-metadata-url https://hornan7.github.io/proposal.txt \
--drep-metadata-hash kjhfdsgb8967fdvtgbv34807bnvs9834nbgv539 \
--out-file drep.cert

following column. (--drep-metadata-hash)

byroi]		build	(stake-verification-key <string></string>	non-extended-key	extended-verification-key-file <file>verification-key-file <file></file></file>			alonzo-era babbage-era		(mainnet testnet-magic <natural></natural>		(genesis-verification-key-file <file>) [protocol-major-version <natural>protocol-minor-version <natural>]</natural></natural></file>
	cation-key-file <file> ng-key-file <file></file></file>	Dulla	stake-verification-key-file <file> stake-key-hash <hash></hash></file>	convert-byron-key	[password <text>]</text>			conway-era (stake-pool-verification-key <string></string>) [out-file <file>]</file>		[decentralization-parameter <rational>] [extra-entropy <hex> reset-extra-entropy</hex></rational>
kev-hash (payn	 ment-verification-key <string> ment-verification-key-file <file></file></string>		stake-script-file <file>) (mainnet testnet-magic <natural>)</natural></file>		(byron-payment-key-type legacy-byron-payment-key-type byron-genesis-key-type			cold-verification-key-file <file>)</file>	tip	[socket-path <socket_path>] [cardano-mode [epoch-slots <slots>]]</slots></socket_path>] [max-block-header-size <natural>]</natural>
j	file <file>]</file>		[out-file <file>]</file>		byron-genesis-key-type legacy-byron-genesis-key-type byron-genesis-delegate-key-type			<pre>(vrf-verification-key <string></string></pre>		(mainnet testnet-magic <natural>) [out-file <file>]</file></natural>		[max-block-body-size <natural>] [max-tx-size <natural>] [min-fee-constant <lovelace>]</lovelace></natural></natural>
			(stake-verification-key <string> stake-verification-key-file <file></file></string>		legacy-byron-genesis-delegate-key-type)			pool-pledge <lovelace>pool-cost <lovelace></lovelace></lovelace>	stalia wa ala	[socket-path <socket_path>]</socket_path>		[min-fee-linear <natural>] [min-utxo-value <natural>]</natural></natural>
payr	ment-verification-key-file <file> ment-script-file <file></file></file>) [out-file <file>]</file>		<pre>(byron-signing-key-file <file> byron-verification-key-file <file>)</file></file></pre>			pool-margin <rational></rational>(pool-reward-account-verification-key <string></string> pool-reward-account-verification-key-file <file></file>	stake-pools	[cardano-mode [epoch-slots <slots>]] (mainnet testnet-magic <natural>)</natural></slots>		[key-reg-deposit-amt <natural>] [pool-reg-deposit <natural>] [min-pool-cost <natural>]</natural></natural></natural>
	e-verification-key <string> ce-verification-key-file <file></file></string>	registration-certificate	(shelley-era allegra-era		out-file <file></file>) (pool-owner-verification-key <string></string>		[output-json output-text] [out-file <file>]</file>		[number-of-pools <natural>] [pool-retirement-epoch-boundary <epoch_boundary>] [number-of-pools <natural>]</natural></epoch_boundary></natural>
stakı stakı stakı	ke-key-hash <hash> ke-script-file <file></file></hash>		mary-era alonzo-era	convert-byron-genesis-vkey	byron-genesis-verification-key <base64>out-file <file></file></base64>			pool-owner-stake-verification-key-file <file>)</file>	stake-distribution	[socket-path <socket_path>] [cardano-mode</socket_path>		[pool-influence <rational>] [monetary-expansion <rational>]</rational></rational>
j	e-address <address> nnet testnet-magic <natural< th=""><th>N.</th><th> babbage-era conway-era</th><th>againent the land</th><th></th><th></th><th></th><th>[[pool-relay-ipv4 <string>] [pool-relay-ipv6 <string>] pool-relay-port <int></int></string></string></th><th></th><th>[cardano-mode [epoch-slots <slots>]] (mainnet</slots></th><th></th><th>[treasury-expansion <rational>] [utxo-cost-per-word <lovelace>] [price-execution-steps <rational></rational></lovelace></rational></th></natural<></address>	N.	babbage-era conway-era	againent the land				[[pool-relay-ipv4 <string>] [pool-relay-ipv6 <string>] pool-relay-port <int></int></string></string>		[cardano-mode [epoch-slots <slots>]] (mainnet</slots>		[treasury-expansion <rational>] [utxo-cost-per-word <lovelace>] [price-execution-steps <rational></rational></lovelace></rational>
[out-fi	file <file>]</file>		(stake-verification-key <string> stake-verification-key-file <file></file></string>	convert-itn-key	itn-verification-key-file <file></file>			single-host-pool-relay <string> [pool-relay-port <int>]</int></string>		testnet-magic <natural>) [out-file <file>]</file></natural>		price-execution-memory <rational>] [max-tx-execution-units (<int, int="">)]</int,></rational>
	ess <address> file <file>]</file></address>		stake-key-hash <hash> stake-script-file <file> stake-address <address></address></file></hash>		out-file <file></file>			multi-host-pool-relay <string> metadata-url <url></url></string>				[max-block-execution-units (<int, int="">)] [max-value-size <int>] [collateral-percent <int>]</int></int></int,>
) [key-reg-deposit-amt <natural></natural>	convert-itn-extended-key	itn-signing-key-file <file> out-file <file></file></file>			metadata-hash <hash>] (mainnet</hash>	stake-address-info	[socket-path <socket_path>] [cardano-mode</socket_path>		[max-collateral-inputs <int>] [utxo-cost-per-byte <lovelace>]</lovelace></int>
			out-file <file></file>	convert-itn-bip32-key				testnet-magic <natural> out-file <file></file></natural>		address <address> (mainnet</address>		[cost-model-file <file>]</file>
cardano-cl	i text-view	deregistration-certificate	(shelley-era allegra-era	convert for sip32 key	out-file <file></file>			out the Stills		testnet-magic <natural> out-file <file>]</file></natural>	create-genesis-key-	[shelley-era allegra-era
			mary-era alonzo-era babbage-era	convert-cardano-address-key	(cc-cold-key cc-hot-key	deregistratio		[shelley-era		[socket-path <socket_path>]</socket_path>	delegation-certificate	mary-era alonzo-era
ecode-chorin-file	e <file> file <file>]</file></file>		conway-era 		drep-key shelley-payment-key	ueregistratio	n-certificate	allegra-era mary-era	utxo	[cardano-mode [epoch-slots <slots>]] (whole-utxo</slots>		babbage-era] (genesis-verification-key <string></string>
[0ut-11	ille Vitilizi		(stake-verification-key <string></string> stake-verification-key-file <file></file> stake-key-hash <hash></hash>		shelley-stake-key icarus-payment-key byron-payment-key			alonzo-era babbage-era conway-era		(address <address>) (tx-in <tx-in>)</tx-in></address>		genesis-verification-key-file <file> genesis-verification-key-hash <string></string></file>
			stake-script-file <file> stake-address <address></address></file>) signing-key-file <file></file>] (stake-pool-verification-key <string></string>		(mainnet testnet-magic <natural>) [output-json output-text]</natural>) (genesis-delegate-verification-key <string></string>
	1) [key-reg-deposit-amt <natural> out-file <file></file></natural>		out-file <file></file>			cold-verification-key-file <file>) epoch <natural></natural></file>		[out-file <file>]</file>		genesis-delegate-verification-key-file <file> genesis-delegate-verification-key-hash <string>)</string></file>
cardano	-cli ping		out-file <file></file>					out-file <file></file>	ledger-state	[socket-path <socket_path>] [cardano-mode [epoch-slots <slots>]]</slots></socket_path>		(vrf-verification-key <string> vrf-verification-key-file <file></file></string>
		delegation-certificate	allegra-era mary-era			id		(stake-pool-verification-key <string></string>		<pre>(mainnet testnet-magic <natural>) [out-file <file>]</file></natural></pre>		vrf-verification-key-hash <string>) out-file <file></file></string>
((-h ho	ount <count>] lost <host>) (-u unixsock <sc< th=""><th>OCKET>))</th><th> alonzo-era babbage-era</th><th></th><th></th><th>, iu</th><th></th><th> cold-verification-key-file <file> </file></th><th>protocol-state</th><th>[socket-path <socket_path>]</socket_path></th><th>create poll</th><th></th></sc<></host></count>	OCKET>))	alonzo-era babbage-era			, iu		cold-verification-key-file <file> </file>	protocol-state	[socket-path <socket_path>]</socket_path>	create poll	
	ort <port>] nagic <magic>] on]</magic></port>		conway-era (stake-verification-key <string></string>					[output-format <string>] [out-file <file>]</file></string>		<pre>[cardano-mode [epoch-slots <slots>]] (mainnet testnet-magic <natural>) [out-file <file>]</file></natural></slots></pre>	create-poll	(answer <string>) [nonce <uint>]</uint></string>
[-q qu			stake-verification-key-file <file> stake-key-hash <hash></hash></file>			metadata-ha		pool-metadata-file <file></file>		[out-file \FILE>]		out-file <file></file>
			stake-script-file <file> stake-address <address></address></file>			metadata-na.	311	[out-file <file>]</file>	stake-snapshot	[socket-path <socket_path>] [cardano-mode [epoch-slots <slots>]]</slots></socket_path>	answer-poll	poll-file <file> [answer <int>]</int></file>
			(stake-pool-verification-key <string></string>							<pre>(mainnet testnet-magic <natural>) [all-stake-pools</natural></pre>		[out-file <file>]</file>
			cold-verification-key-file <file> stake-pool-id <stake-pool-id></stake-pool-id></file>] [out-file <file>]</file>	verify-poll	poll-file <file></file>
) out-file <file></file>						pool-state	[socket-path <socket_path>]</socket_path>		tx-file <file> [out-file <file>]</file></file>
			cardan	o-cli transaction						[cardano-mode [epoch-slots <slots>]] (mainnet testnet-magic <natural>) (stake-pool-id <stake-pool-id>)</stake-pool-id></natural></slots>	create-mir-certificate	
			Caraari) [out-file <file>]</file>	create iiii certineate	allegra-era mary-era
sign	(tx-b	oody-file <file> tx-file <file>)</file></file>	build	[socket-path <socket_path>]</socket_path>	build-raw		[byron-era		leadership-schedule	[socket-path <socket_path>]</socket_path>		alonzo-era babbage-era
	[ac	ning-key-file <file> ddress <string>]] nnet testnet-magic <natural>]</natural></string></file>		[byron-era shelley-era allegra-era			shelley-era allegra-era mary-era			[cardano-mode [epoch-slots <slots>]] (mainnet</slots>		(reserves treasury
	out-	file <file></file>		mary-era alonzo-era			alonzo-era babbage-era			testnet-magic <natural> </natural>) (stake-address <address>) (reward <lovelace>)</lovelace></address>
witness	tx-b	ody-file <file> ing-key-file <file></file></file>		babbage-era [cardano-mode [epoch-slots <slots>]]</slots>] [script-valid scri (tx-in <tx-in></tx-in>	ipt-invalid]		genesis <file> (stake-pool-verification-key <string></string></file>		out-file <file> stake-addresses</file>
	[ma	dress <string>] innet testnet-magic <natural>] file <file></file></natural></string>		(mainnet testnet-magic <natural>) [script-valid script-invalid]</natural>			[spending-tx-in-re (spending-plutu:			cold-verification-key-file <file> stake-pool-id <stake-pool-id>)</stake-pool-id></file>		transfer-to-treasury transfer-to-rewards
				<pre>[witness-override <word>] (tx-in <tx-in> [spending-tx-in-reference <tx-in></tx-in></tx-in></word></pre>			spending-plutus 			vrf-signing-key-file <file> (current next)</file>		
assemble	tx-bo [witi	ody-file <file> ness-file <file>] file <file></file></file></file>		(spending-plutus-script-v2 spending-plutus-script-v3			spending-refer	ence-tx-in-datum-cbor-file <cbor file=""> rence-tx-in-datum-file <json file=""> rence-tx-in-datum-value <json value=""></json></json></cbor>		[output-json output-text] [out-file <file>]</file>	create-mir-certificate	 [shelley-era allegra-era
) (spending-reference-tx-in-datum-cbor-file <cbc <json="" fil<="" th="" spending-reference-tx-in-datum-file=""><th></th><th></th><th>)</th><th>rence-tx-in-inline-datum-present</th><th>kes-period-info</th><th>[socket-path <socket_path>] [cardano-mode</socket_path></th><th>stake-addresses</th><th> mary-era alonzo-era</th></cbc>)	rence-tx-in-inline-datum-present	kes-period-info	[socket-path <socket_path>] [cardano-mode</socket_path>	stake-addresses	mary-era alonzo-era
submit	[car	cket-path <socket_path>] dano-mode [epoch-slots <slots>]] innet testnet-magic <natural>)</natural></slots></socket_path>		spending-reference-tx-in-datum-value <json \<br=""> spending-reference-tx-in-datum-value <json \<br=""> spending-reference-tx-in-inline-datum-present</json></json>	/ALUE>		spending-refer	ence-tx-in-redeemer-cbor-file <cbor file=""> rence-tx-in-redeemer-file <json file=""> rence-tx-in-redeemer-value <json value=""></json></json></cbor>		[cardano-mode [epoch-slots <slots>]] (mainnet testnet-magic <natural>)</natural></slots>		babbage-era (reserves
		le <file></file>		(spending-reference-tx-in-redeemer-cbor-file <0) spending-referer	nce-tx-in-execution-units (<int, int="">)</int,>		op-cert-file <file> [out-file <file>]</file></file>		treasury) (stake-address ADDRESS)
policyid		ot-file <file></file>		spending-reference-tx-in-redeemer-file <json spending-reference-tx-in-redeemer-value <jsc)</jsc </json 			simple-script-tx- tx-in-script-file <	-in-reference <tx-in> <file></file></tx-in>	tx-mempool	[socket-path <socket_path>] [cardano-mode [epoch-slots <slots>]]</slots></socket_path>		(reward LOVELACE)out-file FILE
calculate-min-fee		ody-file <file></file>		simple-script-tx-in-reference <tx-in> tx-in-script-file <file></file></tx-in>			itx-in-datum-fi			(mainnet testnet-magic <natural>) (info next-tx tx-exists)</natural>		To the Unit of the Control of the Co
	prot witr	cocol-params-file <file> ness-count <natural> on-witness-count <natural>]</natural></natural></file>		(tx-in-datum-cbor-file <cbor file=""> tx-in-datum-file <json file=""></json></cbor>			tx-in-datum-v tx-in-inline-da)	ralue <json value=""> atum-present</json>		[out-file <file>]</file>	create-mir-certificate transfer-to-treasury	[shelley-era allegra-era mary-era
	refe	rence-script-size <int> innet</int>		tx-in-datum-value <json value="">tx-in-inline-datum-present</json>			itx-in-redeeme	er-cbor-file <cbor file=""> er-file <json file=""></json></cbor>	slot-number	<pre>[socket-path <socket_path>] [cardano-mode [epoch-slots <slots>]] (mainnet testnet-magic <natural>)</natural></slots></socket_path></pre>		alonzo-era babbage-era
	j	tnet-magic <natural></natural>) (tx-in-redeemer-cbor-file <cbor file=""> tx-in-redeemer-file <json file=""></json></cbor>			j	er-value <json value=""> -units (<int, int="">)]</int,></json>		TIMESTAMP		Jtransfer <lovelace>out-file <file></file></lovelace>
	[tx-c	n-count <natural>] put-count <natural>]</natural></natural>		tx-in-redeemer-value <json value=""></json>]) [read-only-tx-in-ref	ference <tx-in>]</tx-in>				
txid		oody-file <file> tx-file <file>)</file></file>]) [read-only-tx-in-reference <tx-in>] [required-signer <file> required-signer-hash <</file></tx-in>	HASH>]		[tx-in-collateral <t> [tx-out-return-collateral]</t>	ateral <address value="">]</address>			create-mir-certificate transfer-to-rewards	[shelley-era allegra-era
view	[out	 put-json output-yaml] -file <file>]</file>		[tx-in-collateral <tx-in>] [tx-out-return-collateral <address value="">]</address></tx-in>			[required-signer <f< th=""><th>FILE></th><th>cardano-</th><th>cli genesis</th><th></th><th> mary-era alonzo-era babbage-era</th></f<>	FILE>	cardano-	cli genesis		mary-era alonzo-era babbage-era
		oody-file <file> tx-file <file>)</file></file>		<pre>[tx-total-collateral <integer>] [tx-out <address value=""> [tx-out-datum-hash <hash></hash></address></integer></pre>] [tx-out <address \<br="">[tx-out-datum-ha</address>] transfer <lovelace></lovelace>
hash-script-data	(scr	ipt-data-cbor-file <cbor file=""> ipt-data-file <json file=""></json></cbor>		tx-out-datum-hash-cbor-file <cbor file=""> tx-out-datum-hash-file <json file=""></json></cbor>			tx-out-datum-ha tx-out-datum-ha	ash-cbor-file <cbor file=""> ash-file <json file=""></json></cbor>	key-gen-genesis	verification-key-file <file>signing-key-file <file></file></file>		out-file <file></file>
		ipt-data-nie <json file=""></json>		tx-out-datum-hash-value <json value=""> tx-out-datum-embed-cbor-file <cbor file=""> tx-out-datum-embed-file <json file=""></json></cbor></json>			tx-out-datum-er	ash-value <json value=""> mbed-cbor-file <cbor file=""> mbed-file <json file=""></json></cbor></json>	key-gen-delegate	verification-key-file <file></file>	. <u>-</u>	
				tx-out-datum-embed-value <json value=""> tx-out-inline-datum-cbor-file <cbor file=""></cbor></json>			tx-out-datum-er tx-out-inline-dat	mbed-value <json value=""> tum-cbor-file <cbor file=""></cbor></json>		signing-key-file <file>operational-certificate-issue-counter-file <file< th=""><th></th><th></th></file<></file>		
calculate-min-req	she	on-era elley-era egra-era		tx-out-inline-datum-file <json file=""> tx-out-inline-datum-value <json value="">]</json></json>				tum-file <json file=""> tum-value <json value=""></json></json>	key-gen-utxo		card	ano-cli node
	ma alo	nry-era onzo-era		[tx-out-reference-script-file <file>]]change-address <address></address></file>			I [tx-out-reference- [mint <value></value>		key-hash		key-gen	[key-output-format <string>]</string>
		bbage-era nway-era		[mint <value> (mint-script-file <file> [mint-redeemer-cbor-file <cbor file=""></cbor></file></value>			(mint-script-file < [(mint-redeeme	<pre><file> er-cbor-file <cbor file=""></cbor></file></pre>	key-hash	·		cold-verification-key-file <file>cold-signing-key-file <file>operational-certificate-issue-counter-file <file></file></file></file>
	tx-oı	ocol-params-file <file> ut <address value=""></address></file>		mint-redeemer-file <json file=""> mint-redeemer-value <json value=""></json></json>			imint-redeeme	er-cbor-file <cbor file=""> er-file <json file=""> er-value <json value=""></json></json></cbor>	get-ver-key	verification-key-file <file>signing-key-file <file></file></file>		
	tx-	out-datum-hash <hash> out-datum-hash-cbor-file <cbor file=""> out-datum-hash-file <json file=""></json></cbor></hash>] simple-minting-script-tx-in-reference <tx-in> policy-id <hash></hash></tx-in>) mint-execution	-units (<int, int="">)]</int,>	initial-addr	verification-key-file FILE (mainnet testnet-magic NATURAL)	key-gen-KES	[key-output-format <string>]verification-key-file <file>signing-key-file <file></file></file></string>
	tx- tx-	out-datum-hash-value <json value=""> out-datum-embed-cbor-file <cbor file=""></cbor></json>		mint-tx-in-reference <tx-in> (mint-plutus-script-v2</tx-in>			simple-minting-: policy-id <hash mint-tx-in-refere</hash 			[out-file FILE]	kev-gen-VRF	
	tx-	out-datum-embed-file <json file=""> out-datum-embed-value <json value=""> out-inline-datum-cbor-file <cbor file=""></cbor></json></json>		mint-plutus-script-v3) (mint-reference-tx-in-redeemer-cbor-file <cbof< th=""><th>R FII F></th><th></th><th>(mint-plutus-scri mint-plutus-scr</th><th>ipt-v2</th><th>initial-txin</th><th>verification-key-file FILE (mainnet testnet-magic NATURAL)</th><th> Key Sch VIII</th><th>verification-key-file <file>signing-key-file <file></file></file></th></cbof<>	R FII F>		(mint-plutus-scri mint-plutus-scr	ipt-v2	initial-txin	verification-key-file FILE (mainnet testnet-magic NATURAL)	Key Sch VIII	verification-key-file <file>signing-key-file <file></file></file>
	tx-	out-inline-datum-file <json file=""> out-inline-datum-file <json value=""></json></json>		mint-reference-tx-in-redeemer-cbor-file <cbof mint-reference-tx-in-redeemer-file <json file:<br=""> mint-reference-tx-in-redeemer-value <json th="" va<=""><th>></th><th></th><th></th><th>e-tx-in-redeemer-cbor-file <cbor file=""> e-tx-in-redeemer-file <json file=""></json></cbor></th><th></th><th>[out-file FILE]</th><th>key-hash-VRF</th><th>(verification-key <string></string></th></json></json></cbof 	>			e-tx-in-redeemer-cbor-file <cbor file=""> e-tx-in-redeemer-file <json file=""></json></cbor>		[out-file FILE]	key-hash-VRF	(verification-key <string></string>
] [tx-o	out-reference-script-file <file>]</file>) policy-id <hash></hash>			mint-reference 	e-tx-in-redeemer-value <json value=""></json>	create-cardano	genesis-dir <dir> [gen-genesis-keys <int>]</int></dir>		verification-key-file <file>) [out-file <file>]</file></file>
)] [invalid-before <slot>] [invalid-hereafter <slot>]</slot></slot>			mint-reference-t policy-id <hash>)]</hash>	tx-in-execution-units (<int, int="">) ></int,>		[gen-utxo-keys <int>] [start-time <utc-time>]</utc-time></int>	now counter	
				[certificate-file <certificatefile> [certificate-script-file <file></file></certificatefile>			[invalid-before <sl [invalid-hereafter <</sl 			<pre>[supply <lovelace>] [security-param <int>] [slot-length <int>]</int></int></lovelace></pre>	new-counter	(stake-pool-verification-key <string></string> genesis-delegate-verification-key <string></string> cold-verification-key-file <file></file>
				[certificate-redeemer-cbor-file <cbor file=""> certificate-redeemer-file <json file=""> certificate-redeemer-value <json value=""></json></json></cbor>			[fee <lovelace>] [certificate-file <ce [certificate-script<="" th=""><th></th><th></th><th>[slot-coefficient <rational>] (mainnet</rational></th><th></th><th>) counter-value <int></int></th></ce></lovelace>			[slot-coefficient <rational>] (mainnet</rational>) counter-value <int></int>
				;] certificate-tx-in-reference <tx-in></tx-in>			[(certificate-red	leemer-cbor-file <cbor file=""></cbor>		testnet-magic <natural>) byron-template <filepath></filepath></natural>		operational-certificate-issue-counter-file <file></file>
				(certificate-plutus-script-v2 certificate-plutus-script-v3				deemer-file <json file=""> deemer-value <json value=""></json></json>		shelley-template <filepath>alonzo-template <filepath></filepath></filepath>	issue-op-cert	(kes-verification-key <string> kes-verification-key-file <file></file></string>

[--withdrawal <WITHDRAWAL>
[--withdrawal-script-file <FILE>

[--json-metadata-no-schema |--json-metadata-detailed-schema

--withdrawal-execution-units (<INT, INT>)]

| --withdrawal-tx-in-reference <TX-IN>
(--withdrawal-plutus-script-v2
|--withdrawal-plutus-script-v3

(--withdrawal-reference-tx-in-redeemer-cbor-file <CBOR FILE> | --withdrawal-reference-tx-in-redeemer-file <JSON FILE> | --withdrawal-reference-tx-in-redeemer-value <JSON VALUE>)

--withdrawal-reference-tx-in-execution-units (<INT, INT>)

(--certificate-reference-tx-in-redeemer-cbor-file <CBOR FILE> | --certificate-reference-tx-in-redeemer-file <JSON FILE> | --certificate-reference-tx-in-redeemer-value <JSON VALUE>

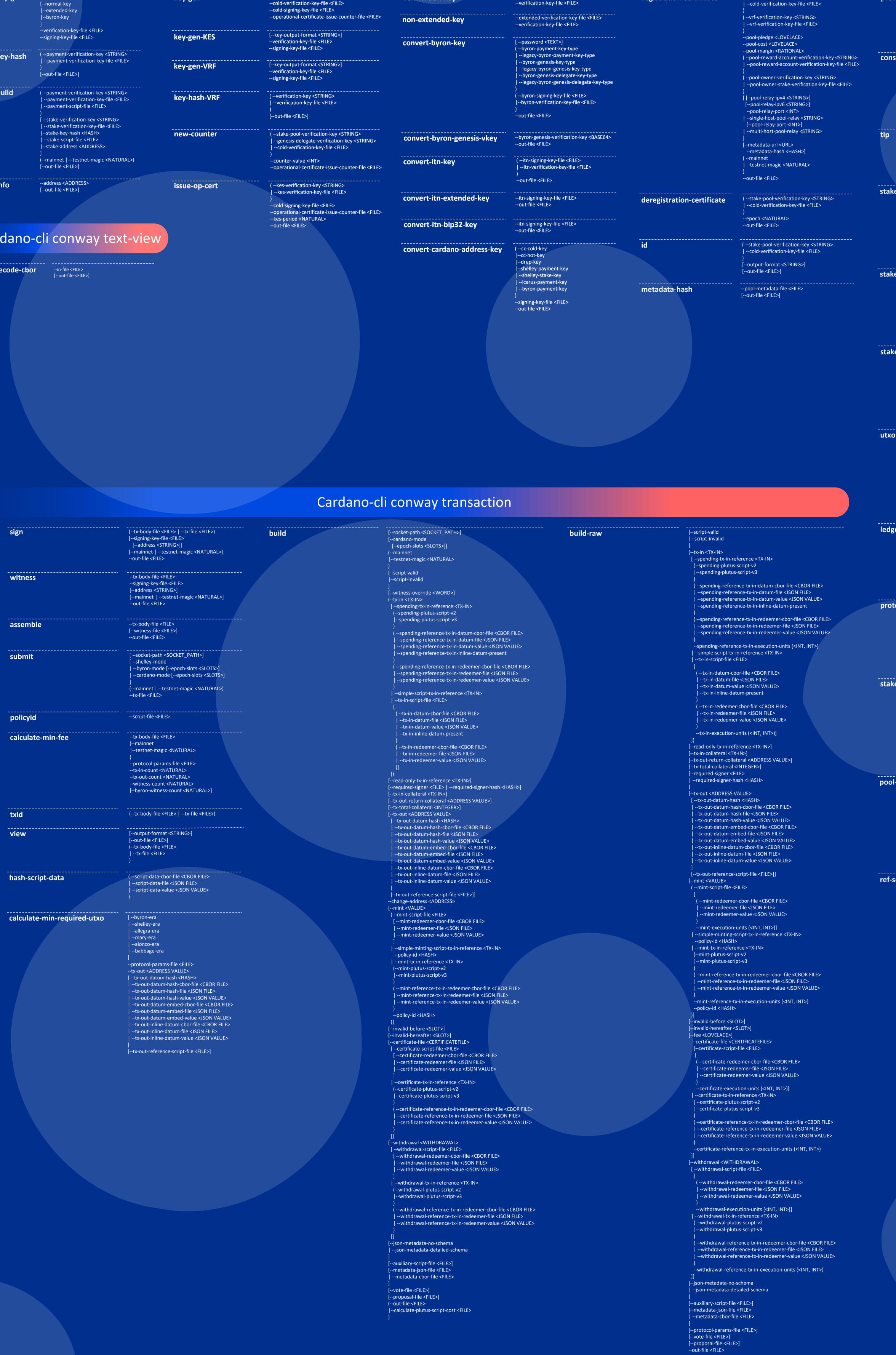
(--withdrawal-reference-tx-in-redeemer-cbor-file <CBOR FILE> | --withdrawal-reference-tx-in-redeemer-file <JSON FILE> | --withdrawal-reference-tx-in-redeemer-value <JSON VALUE>

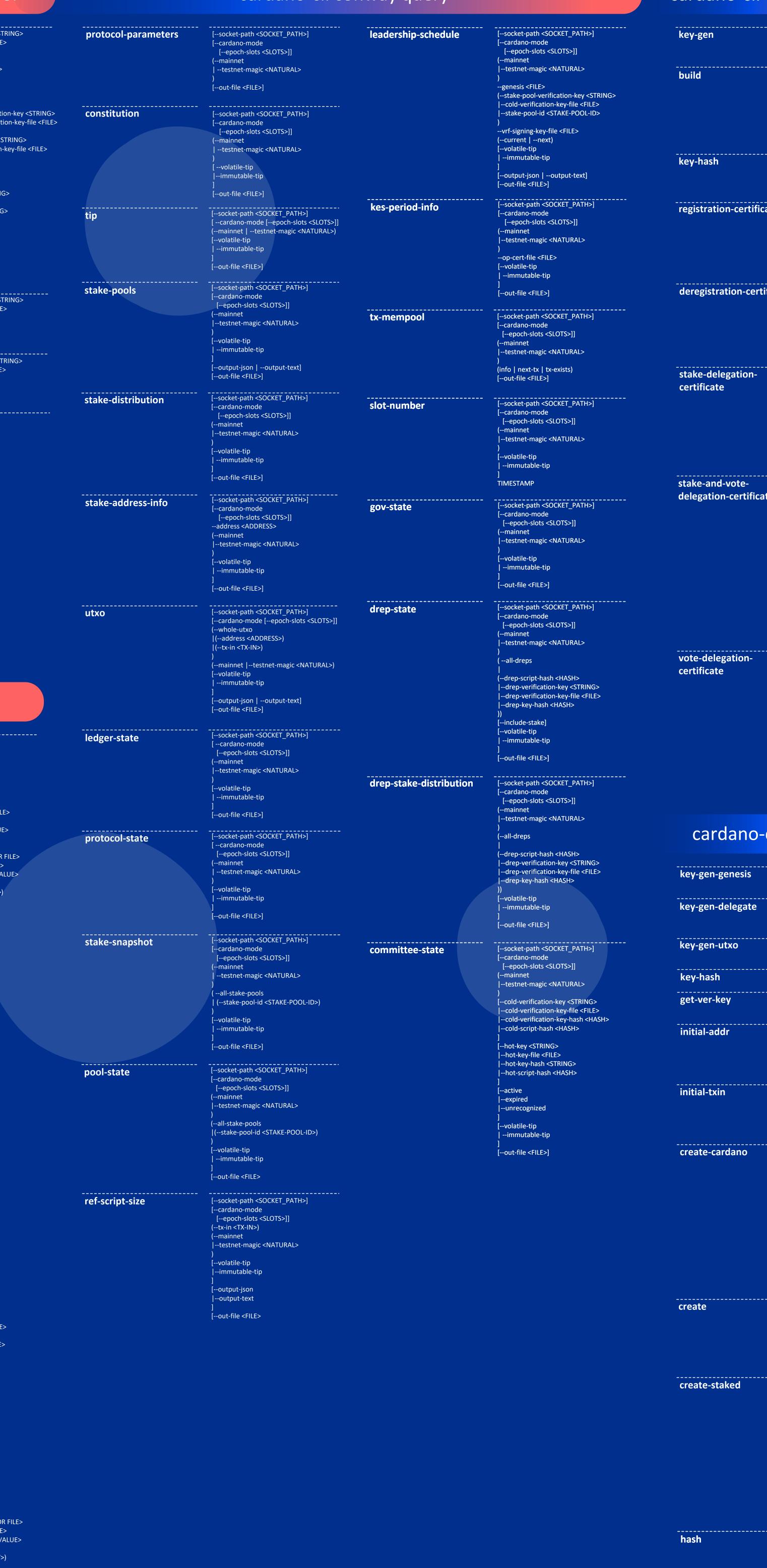
]] [--json-metadata-no-schema | --json-metadata-detailed-schema

cardano-cli query

[--supply <LOVELACE>]
(--mainnet | --testnet-magic <NATURAL>)

ernance	cardano-	cli conway addres	s card
E> FURAL> rification-key-file <file>) rajor-version <natural></natural></file>	key-gen	[key-output-format <string>] [normal-key extended-key byron-key</string>	key-gen
ninor-version <natural>] ation-parameter <rational>] opy <hex> ra-entropy</hex></rational></natural>	 kov bash	verification-key-file <file>signing-key-file <file> (payment-verification-key <string></string></file></file>	key-gen-KE
header-size <natural>] body-size <natural>] ! <natural>] nstant <lovelace>] ear <natural>]</natural></lovelace></natural></natural></natural>	key-hash	payment-verification-key-file <file> out-file <file>]</file></file>	key-gen-VR
alue <natural>] posit-amt <natural>] posit <natural>] ost <natural>] ment-epoch-boundary <epoch_boundary>]</epoch_boundary></natural></natural></natural></natural>	build	(payment-verification-key <string> payment-verification-key-file <file> payment-script-file <file>)</file></file></string>	key-hash-V
-pools <natural>] nce <rational>] expansion <rational>] expansion <rational>] expansion <rational>] per-word <lovelace>] ution-steps <rational> cution-memory <rational>]</rational></rational></lovelace></rational></rational></rational></rational></natural>		[stake-verification-key <string> stake-verification-key-file <file> stake-key-hash <hash> stake-script-file <file> stake-address <address> (mainnet testnet-magic <natural>) [out-file <file>]</file></natural></address></file></hash></file></string>	new-count
cution-units (<int, int="">)] execution-units (<int, int="">)] size <int>] percent <int>] eral-inputs <int>]</int></int></int></int,></int,>	info	address <address> [out-file <file>]</file></address>	issue-op-ce
per-byte <lovelace>] -file <file>]</file></lovelace>			
a a era	cardano-	cli conway text-vie	
erification-key <string> erification-key-file <file> erification-key-hash <string> elegate-verification-key <string> elegate-verification-key-file <file> elegate-verification-key-hash <string> elegate-verification-key-hash <string> extion-key <string> extion-key-file <file> extion-key-file <file> extion-key-hash <string> extion-key-hash <string> extion-key-hash <string></string></string></string></file></file></string></string></string></file></string></string></file></string>	decode-cbor	in-file <file> [out-file <file>]</file></file>	
TRING>) NT>] E> E> T>]			
E>] 			
ra ra			
a era	 sign	(tx-bod	y-file <file> tx-file <file< td=""></file<></file>
ress <address>) OVELACE>) LE> sses</address>		[addr	r-key-file <file> ess <string>]] et testnet-magic <natuf <file></file></natuf </string></file>
reasury rewards	witness	signing [addre:	v-file <file> -key-file <file> ss <string>] et testnet-magic <natu <file></file></natu </string></file></file>
ı Pra	assembl		r-file <file> s-file <file>] <file></file></file></file>
ess ADDRESS) /ELACE)	submit	[shelle byror carda] (mainn	n-mode [epoch-slots <slot no-mode [epoch-slots <sl et testnet-magic <natu< td=""></natu<></sl </slot
	policyid	tx-file < script-f	:FILE> ile <file></file>
ra VELACE> E>	calculate	(mainn testna) protoc tx-in-ca	y-file <file> et et-magic <natural> ol-params-file <file> ount <natural> count <natural></natural></natural></file></natural></file>
ra VVELACE>		[byron	s-count <natural> -witness-count <natural>]</natural></natural>
E>	txid view	[outpu	y-file <file> tx-file <file </file </file>
node	hash-scr	script	data-cbor-file <cbor file=""> data-file <json file=""> data-value <json value=""></json></json></cbor>
put-format <string>] ification-key-file <file> ning-key-file <file> nal-certificate-issue-counter-file <file></file></file></file></string>	calculate	e-min-required-utxo [byron-	y-era
put-format <string>] on-key-file <file> cey-file <file></file></file></string>		allegra mary- alonzo babba 	era o-era ge-era
put-format <string>] on-key-file <file> cey-file <file></file></file></string>		tx-out < [tx-out tx-out	l-params-file <file> ADDRESS VALUE> datum-hash <hash> -datum-hash-cbor-file <cbc -datum-hash-file <json file<="" td=""></json></cbc </hash></file>
ation-key <string> ation-key-file <file> <file>] pool-verification-key <string></string></file></file></string>		tx-out tx-out tx-out tx-out tx-out	-datum-hash-value <json v<br="">-datum-embed-cbor-file <cl -datum-embed-file <json f<br="">-datum-embed-value <json -inline-datum-cbor-file <cbo -inline-datum-file <json fil<="" td=""></json></cbo </json </json></cl </json>
s-delegate-verification-key <string> erification-key-file <file> -value <int> onal-certificate-issue-counter-file <file></file></int></file></string>		i	-inline-datum-value <json <file="" \="" reference-script-file="">]</json>
rification-key <string> rification-key-file <file></file></string>			
ning-key-file <file> onal-certificate-issue-counter-file <file> od <natural> <file></file></natural></file></file>			





socket-path <socket_path>] cardano-mode</socket_path>	key-gen	[key-output-format <string>]verification-key-file <file></file></string>
[epoch-slots <slots>]]mainnettestnet-magic <natural></natural></slots>	build	signing-key-file <file>(stake-verification-key <string></string></file>
nesis <file> ake-pool-verification-key <string> old-verification-key-file <file> ake-pool-id <stake-pool-id></stake-pool-id></file></string></file>		stake-verification-key-file <file> stake-key-hash <hash> stake-script-file <file>)</file></hash></file>
rigning-key-file <file> rent next) atile-tip</file>		(mainnet testnet-magic <natural>) [out-file <file>]</file></natural>
nutable-tip ut-json output-text] file <file>]</file>	key-hash	(stake-verification-key <string> stake-verification-key-file <file>)</file></string>
t-path <socket_path>] no-mode</socket_path>	registration-certificate	[out-file <file>] (stake-verification-key <string> stake-verification-key-file <file></file></string></file>
noch-slots <slots>]] nnet net-magic <natural></natural></slots>		stake-script-file <file> stake-address <address>) [key-reg-deposit-amt <natural></natural></address></file>
ert-file <file> tile-tip nutable-tip</file>	dorogistration contification	out-file <file></file>
ile <file>] et-path <socket_path>] ano-mode</socket_path></file>	deregistration-certificate	(stake-verification-key <string></string> stake-verification-key-file <file></file> stake-script-file <file></file> stake-address <address></address>
no-mode h-slots <slots>]] et et-magic <natural></natural></slots>) [key-reg-deposit-amt <natural> out-file <file></file></natural>
next-tx tx-exists) ile <file>]</file>	stake-delegation- certificate	(stake-verification-key <string> stake-verification-key-file <file> stake-key-hash <hash></hash></file></string>
t-path <socket_path>] no-mode ch-slots <slots>]] net</slots></socket_path>		stake-script-file <file> stake-address <address> (stake-pool-verification-key <string></string></address></file>
magic <natural></natural>		(stake-pool-verification-key < STRING> cold-verification-key-file < FILE> stake-pool-id < STAKE-POOL-ID>) out-file < FILE>
ible-tip	stake-and-vote- delegation-certificate	 (stake-verification-key <string> stake-verification-key-file <file></file></string>
:-path <socket_path>] no-mode ch-slots <slots>]] et</slots></socket_path>	delegation-tertificate	stake-key-hash <hash> stake-script-file <file> stake-address <address>)</address></file></hash>
net et-magic <natural> le-tip</natural>		, (stake-pool-verification-key <string> cold-verification-key-file <file> stake-pool-id <stake-pool-id>)</stake-pool-id></file></string>
table-tip e <file>]</file>		/ (drep-script-hash <hash> drep-verification-key <string> drep-verification-key-file <file> drep-key-hash <hash></hash></file></string></hash>
t-path <socket_path>] no-mode ch-slots <slots>]] net</slots></socket_path>		always-abstain always-no-confidence out-file <file></file>
c <natural></natural>	vote-delegation-	 (stake-verification-key <string> stake-verification-key-file <file></file></string>
sh <hash> on-key <string> on-key-file <file></file></string></hash>	certificate	stake-key-hash <hash> stake-script-file <file> stake-address <address>)</address></file></hash>
h <hash>] ip</hash>) (drep-script-hash <hash> drep-verification-key <string> drep-verification-key-file <file> drep-key-hash <hash></hash></file></string></hash>
		drep-key-nash <hash> always-abstain always-no-confidence) out-file <file></file></hash>
SOCKET_PATH>] e <slots>]]</slots>		out file \File>
c <natural></natural>	cardano-cli c	onway genesis
ash <hash> tion-key <string> tion-key-file <file> h <hash></hash></file></string></hash>	key-gen-genesis	
n <hash></hash>	key-gen-delegate	signing-key-file <file></file>
E>] <socket_path>]</socket_path>	key-gen-utxo	operational-certificate-issue-counter-file <fileverification-key-file <file=""></fileverification-key-file>
-mode slots <slots>]]</slots>	key-hash	signing-key-file <file>verification-key-file <file></file></file>
verification-key <string> -verification-key-file <file> -verification-key-bash <hash></hash></file></string>	get-ver-key	
-verification-key-hash <hash> -script-hash <hash> key <string></string></hash></hash>	initial-addr	verification-key-file FILE (mainnet testnet-magic NATURAL
y-file <file> y-hash <string> ript-hash <hash></hash></string></file>) [out-file FILE]
d ognized	initial-txin	verification-key-file FILE (mainnet testnet-magic NATURAL)
p ole-tip 	 create-cardano	out-file FILE] genesis-dir <dir></dir>
	or sate-saraano	[gen-genesis-keys <int>] [gen-utxo-keys <int>] [start-time <utc-time>] [supply <lovelace>]</lovelace></utc-time></int></int>
		[supply <lovelace>] [security-param <int>] [slot-length <int>] [slot-coefficient <rational>] (mainnet</rational></int></int></lovelace>
		(mainnet testnet-magic <natural>) byron-template <filepath> shelley-template <filepath></filepath></filepath></natural>
		shelley-template <filepath>alonzo-template <filepath>conway-template <filepath> [node-config-template <filepath>]</filepath></filepath></filepath></filepath>
	create	[key-output-format <string>]genesis-dir <dir> [gen-genesis-keys <int>]</int></dir></string>
		[gen-utxo-keys <int>] [start-time <utc-time>] [supply <lovelace>] (mainnet testnet-magic <natural>)</natural></lovelace></utc-time></int>
	create-staked	[key-output-format <string>]genesis-dir <dir></dir></string>
		[gen-genesis-keys <int>] [gen-utxo-keys <int>] [gen-pools <int>]</int></int></int>
		[gen-stake-delegs <int>] [start-time <utc-time>] [supply <lovelace>] [supply-delegated <lovelace>] (main_not</lovelace></lovelace></utc-time></int>
		(mainnet testnet-magic <natural>) [bulk-pool-cred-files <int>]</int></natural>
		[bulk-pools-per-file <int>] [num-stuffed-utxo <int>] [relay-specification-file <file>]</file></int></int>
	hash	genesis <file></file>

	cardano-cli conv	way stake-address	cardano
	key-gen	[key-output-format <string>]verification-key-file <file>signing-key-file <file></file></file></string>	create-constit
NG>	build	(stake-verification-key <string> stake-verification-key-file <file> stake-key-hash <hash> stake-script-file <file>) (mainnet testnet-magic <natural>) [out-file <file>]</file></natural></file></hash></file></string>	
	key-hash	(stake-verification-key <string> stake-verification-key-file <file>) [out-file <file>]</file></file></string>	
	registration-certificate	(stake-verification-key <string> stake-verification-key-file <file> stake-script-file <file> stake-address <address>) [key-reg-deposit-amt <natural> out-file <file></file></natural></address></file></file></string>	update-comm
	deregistration-certificate	(stake-verification-key <string> stake-verification-key-file <file> stake-script-file <file> stake-address <address>) [key-reg-deposit-amt <natural> out-file <file></file></natural></address></file></file></string>	
	stake-delegation- certificate	(stake-verification-key <string> stake-verification-key-file <file> stake-key-hash <hash> stake-script-file <file> stake-address <address>) (stake-pool-verification-key <string> cold-verification-key-file <file></file></string></address></file></hash></file></string>	
	stake-and-vote- delegation-certificate	stake-pool-id <stake-pool-id>)out-file <file>stake-verification-key <string> stake-verification-key-file <file> stake-key-hash <hash> stake-script-file <file> stake-address <address>) (stake-pool-verification-key <string></string></address></file></hash></file></string></file></stake-pool-id>	create-info
		cold-verification-key-file <file> stake-pool-id <stake-pool-id>) (drep-script-hash <hash> drep-verification-key <string> drep-verification-key-file <file> drep-key-hash <hash> always-abstain always-no-confidence)</hash></file></string></hash></stake-pool-id></file>	create-no-con
	vote-delegation- certificate	out-file <file> (stake-verification-key <string> stake-verification-key-file <file> stake-key-hash <hash> stake-script-file <file> stake-address <address>) (drep-script-hash <hash> drep-verification-key <string> drep-verification-key-file <file></file></string></hash></address></file></hash></file></string></file>	
		drep-key-hash <hash> always-abstain always-no-confidence) out-file <file></file></hash>	create-protoco parameters-u
	cardano-cli c	onway genesis	
	key-gen-genesis	verification-key-file <file>signing-key-file <file></file></file>	
	key-gen-delegate	verification-key-file <file>signing-key-file <file>operational-certificate-issue-counter-file <file></file></file></file>	
	key-gen-utxo	verification-key-file <file>signing-key-file <file></file></file>	
	key-hash get-ver-key	verification-key-file <file></file>	
1 >	initial-addr	signing-key-file <file>verification-key-file FILE (mainnet testnet-magic NATURAL) [out-file FILE]</file>	
	initial-txin	verification-key-file FILE (mainnet testnet-magic NATURAL) [out-file FILE]	
	create-cardano	genesis-dir <dir> [gen-genesis-keys <int>] [gen-utxo-keys <int>] [start-time <utc-time>] [supply <lovelace>] [security-param <int>] [slot-length <int>] [slot-coefficient <rational>] (mainnet testnet-magic <natural>)byron-template <filepath>shelley-template <filepath>alonzo-template <filepath>conway-template <filepath></filepath></filepath></filepath></filepath></natural></rational></int></int></lovelace></utc-time></int></int></dir>	
	create	[node-config-template <filepath>] [key-output-format <string>]genesis-dir <dir> [gen-genesis-keys <int>] [gen-utxo-keys <int>] [start-time <utc-time>] [supply <lovelace>] (mainnet testnet-magic <natural>)</natural></lovelace></utc-time></int></int></dir></string></filepath>	create-treasui withdrawal
	create-staked	[mainnet testnet-magic <natural>) [key-output-format <string>]genesis-dir <dir> [gen-genesis-keys <int>] [gen-utxo-keys <int>] [gen-pools <int>] [gen-stake-delegs <int>] [start-time <utc-time>] [supply <lovelace>] [supply-delegated <lovelace>] (mainnet testnet-magic <natural>) [bulk-pool-cred-files <int>]</int></natural></lovelace></lovelace></utc-time></int></int></int></int></dir></string></natural>	view
	 hash	[bulk-pool-cred-files < INT>] [bulk-pools-per-file < INT>] [num-stuffed-utxo < INT>] [relay-specification-file < FILE>]genesis < FILE>	

ano-cli conv	way stake-address	cardano-cli conv	way governance action	cardano-cli conw	ay governance committee
n	[key-output-format <string>]verification-key-file <file>signing-key-file <file></file></file></string>	create-constitution	 (mainnet testnet)	key-gen-cold	
	(stake-verification-key <string> stake-verification-key-file <file></file></string>		governance-action-deposit <natural> (deposit-return-stake-verification-key <string> deposit-return-stake-verification-key-file <file> deposit-return-stake-key-hash <hash></hash></file></string></natural>	key-gen-hot	
	stake-verification-key-file <file> stake-key-hash <hash> stake-script-file <file>) (mainnet testnet-magic <natural>) [out-file <file>]</file></natural></file></hash></file>		deposit-return-stake-key-hash <hash> deposit-return-stake-script-file <file> deposit-return-stake-address <address>) [prev-governance-action-tx-id <txid>prev-governance-action-index <word32>]</word32></txid></address></file></hash>	key-hash	(verification-key <string> verification-key-file <file>)</file></string>
sh	(stake-verification-key <string> stake-verification-key-file <file>) [out-file <file>]</file></file></string>		anchor-url <text>anchor-data-hash <hash>constitution-url <text>constitution-hash <hash> [constitution-script-hash <hash>]out-file <file></file></hash></hash></text></hash></text>	create-hot-key- authorization-certificate	(cold-verification-key <string> cold-verification-key-file <file> cold-verification-key-hash <string> cold-script-file <file>)</file></string></file></string>
ation-certificate	(stake-verification-key <string> stake-verification-key-file <file> stake-script-file <file> stake-address <address></address></file></file></string>	update-committee			(hot-key <string> hot-key-file <file> hot-key-hash <string> hot-script-file <file>) out-file <file></file></file></string></file></string>
) [key-reg-deposit-amt <natural> out-file <file></file></natural>		governance-action-deposit <natural> (deposit-return-stake-verification-key <string> deposit-return-stake-verification-key-file <file> deposit-return-stake-key-hash <hash> deposit-return-stake-script-file <file> deposit-return-stake-address <address></address></file></hash></file></string></natural>	create-cold-key- resignation-certificate	(cold-verification-key <string> cold-verification-key-file <file> cold-verification-key-hash <string> cold-script-hash <hash></hash></string></file></string>
stration-certificate	 (stake-verification-key <string></string> stake-verification-key-file <file></file> stake-script-file <file></file> stake-address <address></address>) [key-reg-deposit-amt <natural></natural>)anchor-url <text>anchor-data-hash <hash> [remove-cc-cold-verification-key <string> remove-cc-cold-verification-key-file <file> remove-cc-cold-verification-key-hash <string></string></file></string></hash></text>) [resignation-metadata-url <text> resignation-metadata-hash <hash>out-file <file></file></hash></text>
	out-file <file></file>		remove-cc-cold-script-hash <hash> </hash>	cardano-cli cor	nway governance drep
delegation- ate	(stake-verification-key <string></string> stake-verification-key-file <file></file> stake-key-hash <hash></hash> stake-script-file <file></file> stake-address <address></address>)		<pre>(add-cc-cold-verification-key <string></string></pre>	key-gen	
	(stake-pool-verification-key <string> cold-verification-key-file <file> stake-pool-id <stake-pool-id>) out-file <file></file></stake-pool-id></file></string>		threshold <rational> [prev-governance-action-tx-id <txid>prev-governance-action-index <word32>]out-file <file></file></word32></txid></rational>	id	(drep-verification-key <string></string> drep-verification-key-file <file></file>)[output-format <string>]</string>[out-file <file>]</file>
ind-vote- tion-certificate	(stake-verification-key <string> stake-verification-key-file <file> stake-key-hash <hash> stake-script-file <file></file></hash></file></string>	create-info	(mainnet testnet)governance-action-deposit <natural> (deposit-return-stake-verification-key <string> deposit-return-stake-verification-key-file <file> deposit-return-stake-key-hash <hash></hash></file></string></natural>	registration-certificate	(drep-script-hash <hash> drep-verification-key <string> drep-verification-key-file <file></file></string></hash>
	stake-address <address>) (stake-pool-verification-key <string> cold-verification-key-file <file> stake-pool-id <stake-pool-id>)</stake-pool-id></file></string></address>		deposit-return-stake-key-hash <hash> deposit-return-stake-script-file <file> deposit-return-stake-address <address>) anchor-url <text> anchor-data-hash <hash></hash></text></address></file></hash>		drep-key-hash <hash>)key-reg-deposit-amt <natural> [drep-metadata-url <text>drep-metadata-hash <hash>]out-file <file></file></hash></text></natural></hash>
	(drep-script-hash <hash> drep-verification-key <string> drep-verification-key-file <file> drep-key-hash <hash> always-abstain</hash></file></string></hash>	create-no-confidence	out-file <file> (mainnet testnet)</file>	retirement-certificate	out-file <file></file>
 elegation-	always-no-confidence out-file <file> (stake-verification-key <string></string></file>)governance-action-deposit <natural> (deposit-return-stake-verification-key <string> deposit-return-stake-verification-key-file <file> deposit-return-stake-key-hash <hash> deposit-return-stake-script-file <file></file></hash></file></string></natural>	metadata-hash)deposit-amt <lovelace>out-file <file>deposit-amt <lovelace>out-file <file></file></lovelace></file></lovelace>
elegation- ate	stake-verification-key-file <file> stake-key-hash <hash> stake-script-file <file> stake-address <address>)</address></file></hash></file>		deposit-return-stake-address < ADDRESS>)anchor-url < TEXT>anchor-data-hash < HASH>governance-action-tx-id < TXID>		[out-file <file>]</file>
	(drep-script-hash <hash> drep-verification-key <string> drep-verification-key-file <file> drep-key-hash <hash> always-abstain</hash></file></string></hash>	create-protocol-	governance-action-index <word32>out-file <file> (mainnet testnet</file></word32>	cardano-cli co	nway governance vote
	always-no-confidence out-file <file></file>	parameters-update)governance-action-deposit <natural> (deposit-return-stake-verification-key <string> deposit-return-stake-verification-key-file <file> deposit-return-stake-key-hash <hash></hash></file></string></natural>	create	(yes no abstain)governance-action-tx-id <txid></txid>governance-action-index <word32></word32>(drep-verification-key <string></string> drep-verification-key-file <file></file>
rdano-cli co	onway genesis		deposit-return-stake-script-file <file> deposit-return-stake-address <address>) anchor-url <text> anchor-data-hash <hash> [prev-governance-action-tx-id <txid></txid></hash></text></address></file>		drep-key-hash <hash> drep-script-hash <hash> stake-pool-verification-key <string> cold-verification-key-file <file> stake-pool-id <stake_pool_id> cc-hot-verification-key <string></string></stake_pool_id></file></string></hash></hash>
n-genesis	verification-key-file <file></file>		prev-governance-action-tx-ld < TXID>prev-governance-action-index < WORD32>] [constitution-script-hash < HASH>] [min-fee-linear < LOVELACE>]		cc-hot-verification-key-file <file> cc-hot-key-hash <string> cc-hot-script-hash <hash></hash></string></file>
n-delegate	signing-key-file <file>verification-key-file <file></file></file>		[min-fee-constant <lovelace>] [max-block-body-size <word32>] [max-tx-size <word32>]</word32></word32></lovelace>) [anchor-url <text> anchor-data-hash <hash>]</hash></text>
n-uelegate n-utxo	signing-key-file <file>operational-certificate-issue-counter-file <file>verification-key-file <file>signing-key-file <file></file></file></file></file>		[max-block-header-size <word16>] [key-reg-deposit-amt <natural>] [pool-reg-deposit <natural>] [pool-retirement-epoch-interval <word32>] [number-of-pools <natural>] [pool-influence <rational>]</rational></natural></word32></natural></natural></word16>	view	out-file <file> [output-json output-yaml]vote-file <file> [out-file <file>]</file></file></file>
r-key	verification-key-file <file>verification-key-file <file>signing-key-file <file></file></file></file>		[treasury-expansion <rational>] [monetary-expansion <rational>] [min-pool-cost <natural>] [price-execution-steps <rational>price-execution-memory <rational>]</rational></rational></natural></rational></rational>	cardano-cli cor	nway governance hash
addr	signing-key-file <file>verification-key-file FILE (mainnet testnet-magic NATURAL</file>		[max-tx-execution-units <int, int="">] [max-block-execution-units <int, int="">] [max-value-size <int>] [collateral-percent <int>] [max-collateral-inputs <int>] [utxo-cost-per-byte <lovelace>]</lovelace></int></int></int></int,></int,>	anchor-data	(text <text> file-binary <file></file></text>
txin) [out-file FILE] verification-key-file FILE (mainnet		[pool-voting-threshold-motion-no-confidence <rational>pool-voting-treshold-committee-normal <rational>pool-voting-treshold-committee-no-confidence <rational>pool-voting-threshold-hard-fork-initiation <rational>pool-voting-threshold-pp-security-group <rational>]</rational></rational></rational></rational></rational>	script	file-text <file>) [out-file <file>]script-file <file></file></file></file>
cardana	testnet-magic NATURAL out-file FILE]		[drep-voting-threshold-motion-no-confidence <rational>drep-voting-threshold-committee-normal <rational>drep-voting-threshold-committee-no-confidence <rational>drep-voting-threshold-update-to-constitution <rational>drep-voting-threshold-hard-fork-initiation <rational></rational></rational></rational></rational></rational>		[out-file <file>]</file>
-cardano	genesis-dir <dir> [gen-genesis-keys <int>] [gen-utxo-keys <int>] [start-time <utc-time>] [supply <lovelace>] [security-param <int>]</int></lovelace></utc-time></int></int></dir>		 drep-voting-threshold-pp-network-group <rational></rational> drep-voting-threshold-pp-economic-group <rational></rational> drep-voting-threshold-pp-technical-group <rational></rational> drep-voting-threshold-pp-governance-group <rational></rational> drep-voting-threshold-treasury-withdrawal <rational>]</rational> [min-committee-size <int>]</int> 		
	[slot-length <int>] [slot-coefficient <rational>] (mainnet testnet-magic <natural>)</natural></rational></int>		[committee-term-length <word32>] [governance-action-lifetime <word32>] [new-governance-action-deposit <natural>] [drep-deposit <lovelace>] [drep-activity <word32>]</word32></lovelace></natural></word32></word32>		
	byron-template <filepath>shelley-template <filepath>alonzo-template <filepath>conway-template <filepath> [node-config-template <filepath>]</filepath></filepath></filepath></filepath></filepath>	create-treasury-	[cost-model-file <file>]out-file <file> (mainnet testnet</file></file>		
	[key-output-format <string>]genesis-dir <dir> [gen-genesis-keys <int>] [gen-utxo-keys <int>] [start-time <utc-time>]</utc-time></int></int></dir></string>	withdrawal)governance-action-deposit <natural> (deposit-return-stake-verification-key <string> deposit-return-stake-verification-key-file <file> deposit-return-stake-key-hash <hash> deposit-return-stake-script-file <file></file></hash></file></string></natural>		
	[supply <lovelace>] (mainnet testnet-magic <natural>)</natural></lovelace>		deposit-return-stake-address <address>)anchor-url <text></text></address>		
-staked	[key-output-format <string>]genesis-dir <dir> [gen-genesis-keys <int>] [gen-utxo-keys <int>] [gen-pools <int>]</int></int></int></dir></string>		anchor-data-hash <hash> [</hash>		
	[gen-stake-delegs <int>]</int>[start-time <utc-time>]</utc-time>[supply <lovelace>]</lovelace>[supply-delegated <lovelace>]</lovelace>(mainnet	 view	action-file <file></file>		
	testnet-magic <natural>) [bulk-pool-cred-files <int>] [bulk-pools-per-file <int>] [num-stuffed-utxo <int>] [relay-specification-file <file>]</file></int></int></int></natural>		[output-json output-yaml] [out-file <file>]</file>		
	genesis <file></file>				

May you enjoy using the CLI as much as I enjoyed building this diagram for the community. Together we will change the world, one command at a time